KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Technology

Computer vision techniques for automatic analysis of mobile eye-tracking data

Stijn DE BEUGHER

Supervisor:
Prof. dr. ir. Toon Goedemé
Prof. dr. Geert Brône, co-supervisor
Prof. dr. ir. Tinne Tuytelaars, co-supervisor

Dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Engineering Technology

November 2016

Computer vision techniques for automatic analysis of mobile eye-tracking data

Stijn DE BEUGHER

Examination committee:
Prof. dr. ir. Boudewijn Meesschaert, chair
Prof. dr. ir. Toon Goedemé, supervisor
Prof. dr. Geert Brône, co-supervisor
Prof. dr. ir. Tinne Tuytelaars, co-supervisor
Prof. dr. Peter De Graef
Prof. dr. ir. Luc Van Eycken
Prof. dr. Joost Vennekens
Dr. Jelle Demanet
Dr. Bert Oben
Dr. Thies Pfeiffer
(University of Bielefeld)

Dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Engineering Technology

November 2016

Preface

When I started this PhD project four years ago, my enthusiasm was sparked from the very first day. Until today, this enthusiasm has not diminished at all. For me, the past four years meant much more than only obtaining an academic degree. It meant a period of intense personal and professional growth. Professionally, I expanded my knowledge of several computer vision techniques and methodologies and I became more experienced in the world of mobile eye-tracking. The countless hours I spent reading, writing and developing software provided me with a critical scientific attitude and a professional maturity which will, without doubt, prove to be of great value throughout the remainder of my career. On a personal level, I got the opportunity to meet several interesting people on the international conferences I attended, with whom I could discuss and explore new insights on my research. Furthermore, on my travels to the United States of America, Portugal, Italy, etc., there was always a little time for some sightseeing and culture-tasting.

None of this would have been possible without the support and help of numerous people, which I sincerely would like to thank.

First of all, I would like to thank my supervisor, Prof dr. ir. Toon Goedemé, for giving me the opportunity to fulfil this PhD project. Toon, sincere thanks for guiding me throughout this journey and for your continued support throughout these past years. Your suggestions and guidance were indispensable and unmistakably contributed to the successful completion of my PhD. Your comments and detailed feedback on my research papers were sometimes a bit overwhelming, but always proved to be invaluable to the quality of my work.

I also would like to thank my co-supervisor Prof. dr. Geert Brône, who was my second mentor throughout this journey. Geert introduced me into the world of mobile eye-tracking and his experience and expertise in this domain was a great help for me to reveal and understand the analytical problems that are related to mobile eye-tracking. Geert, many thanks for your contribution to

each of my scientific publications. Besides offering substantive suggestions, your grammatical suggestions were magnificent and, without doubt, improved the readability of my papers.

Furthermore, I would like to thank my second co-supervisor Prof. dr. ir. Tinne Tuytelaars, as well as each member of my examination committee for their valuable feedback and the time they invested in proofreading this dissertation. Their feedback gave me the opportunity to further enhance this thesis.

Lots of gratitude also go out to my colleagues of the EAVISE research group. Through our numerous interesting discussions at the coffee corner, team-buildings and other non-work related activities, our group of colleagues really became a team. There are a few colleagues in particular that I would like to thank for their contributions to this dissertation. Kristof, thank you for always being available to assist in any of my eye-tracking experiments, even though many considered you as odd when strolling across our campus wearing these strange glasses. You, together with Dries and Steven, always were the first to help me out with several practical and technical issues. Together, we spent countless hours discussing our work and even our personal struggles. Kristof, Steven and Dries, I not only consider you as valuable colleagues, but also as true friends.

Special thanks go to my family and my family-in-law for believing in me and for supporting me. In particular, I would like to thank my mother, for giving me the opportunity to go to college and - even more - to make me the person I am today. Furthermore, I want to express my gratitude to the rest of my family and my family-in-law for the interest they have always shown in my research and the many - much appreciated - leisures they provided me, like family dinners, woodworking and electronics projects.

Finally, I would like to thank my girlfriend Sofie whose support was indispensable. Not only did she spend countless hours of proofreading, she has always been my mainstay in difficult moments. Without doubt, she was partly responsible for successfully finalising this PhD.

Abstract

In the last four decades eye-tracking research has established itself as a powerful paradigm for studying human visual behaviour. More recently, efforts have been made to extend the application field for eye-tracking research beyond the boundaries of lab-based experiments. For example, research on marketing or on human-human interaction definitely benefit from real-life experiments.

Since 1999 the concept of mobile eye-tracking is introduced. A mobile eye-tracker is de facto a sophisticated pair of glasses with a front camera, capturing the field of view, and a second camera which is directed towards the eyes and records the eye movements. Both recordings are combined to determine at which position in the field of view one is looking. The popularity of mobile eye-trackers as a measurement of user experience and behaviour in very diverse application areas is increasing rapidly. Unfortunately, this is tempered by the unfavourable property that a mobile eye-tracker produces a large amount of data that needs to be analysed. The analysis of an eye-tracking experiment can be defined as: 'determine for how long and how often a person looks at a relevant object'. Indeed, depending on the purpose of each eye-tracking experiment, these relevant objects may vary from products on a shelf in the context of market research, up to the face of a person in an experiment on human-human interaction.

In the last decade several attempts have been made to facilitate this analytical challenge. Unfortunately, the existing methods require experimental control and therefore impose restrictions on the concept of real-life mobile eye-tracking. The marker-based analysis, for example, allows for a partial automatic analysis. However, this method confines the flexibility of mobile eye-tracking. Other solutions such as automatic semantic analysis are only applicable for the analysis of a limited range of eye-tracking applications. Therefore, many eye-tracking researchers are often forced to manually analyse the recordings, which is a painstaking and time-consuming task. To overcome these issues, in this dissertation we proposed a computer vision-based framework for the

semi-automatic analysis of mobile eye-tracking recordings.

The goal of this PhD project was to apply computer vision algorithms for the automatic analysis of mobile eye-tracking recordings. By using computer vision algorithms to automatically detect relevant objects in images captured by the scene camera of a mobile eye-tracker we are for example able to automatically determine whether or not a person looked at the objects and how often and for how long one was looking at these relevant objects. Without doubt, efforts to automate this type of analysis can contribute to the increasing popularity of mobile eye-tracking in a broad range of applications.

Developing such an analysis framework is not a trivial task since several challenges need to be tackled. First, it is of vital importance that the accuracy of the analysis is as high as possible. Furthermore, it is advisable that the automatic analysis is faster than manual analysis and even more important, that by using our framework the manual workload significantly decreases. Third, the images that we process are recorded in unconstrained environments using a wearable device. This results in challenging images in which low illumination and motion blur is often present, making the automatic analysis much more complex. Furthermore, we aim to analyse the visual behaviour w.r.t. small moving objects such as the hand gestures of another person, making the analysis even more challenging.

Throughout this PhD project, we focused on four main classes to be recognised. Our analysis framework is capable of analysing the visual behaviour w.r.t. objects (such as specific products in a shopping experiment), human bodies and faces, human hands and gestures. Furthermore, we proposed a semi-automatic analysis approach in which manual intervention and automatic analysis are efficiently intertwined to ensure high accuracy even in challenging conditions.

To fully validate the capabilities of our analysis framework, we recorded a broad range of eye-tracking recordings and used our framework for the validation. This profound validation revealed the applicability of our approach for various types of eye-tracking experiments.

Beknopte samenvatting

In de voorbije veertig jaar heeft eye-tracking zichzelf ontwikkeld als een krachtige methode om menselijk kijkgedrag te analyseren. Traditioneel werden eye-tracking experimenten echter enkel toegepast in beperkte, sterk gecontroleerde omstandigheden. Tegenwoordig wordt er meer aandacht besteed aan het uitbreiden van het applicatiedomein van eye-trackers om meer realistische experimenten toe te laten. Zo hebben bijvoorbeeld onderzoek naar kijk- en koopgedrag van klanten of menselijke interactie zeker baat bij de mogelijkheden van dergelijke realistische experimenten.

In 1999 werd het concept van mobiele eye-tracking geïntroduceerd. Hierbij wordt gebruik gemaakt van een geavanceerde bril waarop meerdere camera's zijn bevestigd. Eén camera is voorwaarts gericht, en filmt dus het gezichtsveld van de persoon die de bril draagt. Een tweede camera wordt naar het oog gericht en filmt de oogbewegingen. Door beide beelden te combineren weet men naar waar de desbetreffende persoon kijkt.

Deze mobiele eye-trackers worden steeds vaker toegepast in zeer diverse domeinen. Hun opmars wordt enkel tegengehouden door het feit dat de toestellen zeer veel en complexe data genereren die moet verwerkt worden. Het verwerken van dergelijke data kan men bijvoorbeeld definiëren als 'bepalen hoe vaak en hoe lang iemand naar een relevant object keek'. Afhankelijk van het doel van het specifieke experiment kunnen de relevante objecten heel divers zijn. In het geval van een marktonderzoek bijvoorbeeld kan het gaan om specifieke producten in een winkelrek. Bij experimenten rond menselijke interactie daarentegen kunnen dan weer de handen of bijvoorbeeld het gezicht van een andere persoon bekeken worden.

In de voorbije tien jaar heeft men heel wat pogingen ondernomen om de analyse van mobiele eye-tracking data te vergemakkelijken. Jammer genoeg leggen bestaande systemen veel eisen op aan de experimenten. Marker-gebaseerde analyse maakt het bijvoorbeeld mogelijk de analyse (deels) te automatiseren,

maar deze techniek beperkt de flexibiliteit van mobiele eye-trackers sterk. Andere oplossingen zoals de recente automatische semantische analyse, zijn slechts toepasbaar voor de analyse van specifieke opnames. Hierdoor zijn veel onderzoekers die werken met data van een mobiele eye-tracker genoodzaakt om de analyse manueel uit te voeren, wat een frustrerende en tijdrovende taak is. In dit doctoraat presenteren wij daarom een alternatieve, computervisie-gebaseerde methodologie om dergelijke opnames automatisch te analyseren.

Het doel van dit doctoraat is dus om computervisietechnieken te gebruiken voor de analyse van mobiele eye-tracker opnames. Door gebruik te maken van dergelijke technieken zijn we in staat om automatisch relevante objecten te detecteren in de beelden die werden opgenomen door de mobiele eye-tracker. Hierdoor zijn we onder andere in staat te bepalen of ze al dan niet werden bekeken gedurende de opname, alsook het vaak en hoe lang men keek naar de betreffende objecten.

Het ontwikkelen van een dergelijk systeem is niet triviaal, en omvat verschillende uitdagingen. We mikken namelijk op een automatisch analyse-systeem dat zo accuraat mogelijk werkt. Ook is het belangrijk dat de automatische analyse sneller verloopt dan een volledig manuele analyse. Verder verwerken we opnames die worden gemaakt in ongecontroleerde omstandigheden en hebben we – omwille van de mobiele eye-tracker – vaak te maken met onscherpe en onderbelichte afbeeldingen, wat de analyse significant bemoeilijkt. Bovendien trachten we het kijkgedrag naar bewegende objecten te analyseren, bijvoorbeeld de gebaren van een andere persoon. Het automatiseren van dergelijke analyses kan zonder twijfel bijdragen aan de stijgende populariteit van mobiele eye-tracking in verscheidene toepassingen.

Doorheen dit doctoraat werden vier grote pijlers uitgewerkt. We hebben systemen ontwikkeld om automatisch het kijkgedrag naar *objecten*, *mensen*, *handen* en *gebaren* te analyseren. Bovendien hebben we een methodologie ontwikkeld waarbij automatische analyse en manuele input optimaal met elkaar werden geïntegreerd om een maximale nauwkeurigheid te behalen, zelfs in extreme omstandigheden.

Om het potentieel van ons systeem ten volle te valideren hebben we verschillende, zeer diverse opnames gemaakt met een mobiele eye-tracker, in uiteenlopende toepassingsgebieden. Onze diepgaande analyses hebben de praktische relevantie en toepasbaarheid van ons systeem bevestigd.

Glossary

ANN Artificial Neural Network. A network inspired by biological neural networks.

AOA Area Of Analysis. Area in which the eye movements are analysed automatically using markers.

AOI Area Of Interest. Area that is relevant for an eye-tracking recording.

AR Augmented Reality. An application of virtual reality in the real world.

ASL American Sign Language. Predominant sign language of deaf communities in the United States.

BRISK Binary Robust Invariant Scalable Keypoints. An algorithm in computer vision to detect and describe local features in images [86].

CNN Convolutional Neural Networks. An object recognition approach based on deep learning.

DPM Deformable Part Model. A pedestrian detection methodology presented in [51].

DTW Dynamic Time Warping. An algorithm for measuring similarities between two temporal sequences that may vary in speed.

EAVISE Embedded Artificially intelligent Vision Engineering.

EEG Electro-EncephaloGraphy. Method to record the activity of the brain.

EOG Electro-OculoGraphy. Technique for measuring the eye movements using the electric potential differences of the skin.

FFLD Fast Fourrier Linear Detector. A faster implementation of the DPM presented in [44].

FN False Negatives. Indicating instances that are unfairly not being classified as the object to be detected.

FPDW Fastest Pedestrian Detector in the West. A pedestrian detection methodology presented in [40].

FPS Frames Per Second. Number of frames that are being processed per second.

FP False Positives. Indicating instances that are unfairly classified as the object to be detected.

GUI Graphical User Interface.

HCI Human Computer Interaction.

HMM Hidden Markov Model. A statistical model.

HOG Histograms Of Oriented Gradients. A pedestrian detection methodology presented in [32].

HSV Hue Saturation Value. A cylindrical color model.

ICF Integral Channel Features. A pedestrian detection methodology presented in [41].

IR Infra-Red. Light that is invisible for the human eye.

KLT Kanade-Lucas-Tomasi. An approach for feature extraction and tracking.

LED Light Emitting Diode. A semiconductor light source.

MIDI Multimodality, Interaction & Discourse.

MLRF Multi-Layered Random Forest. A classification methodology.

MoG Mixtures of Gaussian. A probabilistic model.

NMS Non-Maxima-Suppression. A method for clustering overlapping detection windows.

ORB Oriented FAST and Rotated BRIEF. An algorithm in computer vision to detect and describe local features in images [112].

POG Photo-OculoGraphy. Entails a variety of techniques for eye movements recording involving the measurement of various features.

POR Point Of Regard. The point at which the eye is looking.

RANSAC RANdom SAmple Consensus. An iterative method to estimate parameters of a mathematical model from a set of observed data.

RGB Red Green Blue. An additive color model.

ROI Region Of Interest. Region of the image that we analyse.

SaGA Speech and Gesture Alignment Corpus. A corpus on gestures.

SIFT Scale Invariant Feature Transform. An algorithm in computer vision to detect and describe local features in images [87].

SLAM Simultaneous Localization And Mapping. A methodology of developing and updating a map of an unknown environment.

SVM Support Vector Machine. A classification methodology.

TN True Negatives. Indicating instances that are correctly not being classified as the object to be detected.

ToF Time-of-Flight. A range imaging system that resolves distance based on the speed of light.

TP True Positives. Indicating instances that are correctly classified as the object to be detected.

VJ Viola and Jones. A pedestrian detection methodology presented in [135].

VOG Video-OculoGraphy. Entails a variety of techniques for eye movement recordings involving the measurement of various features.

VR Virtual Reality. A computer technology that simulates an environment with which a user may interact.

YCbCr Luma Chroma blue Chroma red. A color model.

Contents

Abstract iii

Glossary vii

Contents xi

List of Figures xv

List of Tables xxi

1 Introduction 1

1.1 Main contributions 6

1.2 Outline of this dissertation 7

2 (Mobile) eye-tracking 9

2.1 Anatomy of the eye 9

2.2 Eye movements 10

2.3 Eye-tracking techniques 12

2.3.1 Electro-OculoGraphy (EOG) 12

2.3.2 Scleral contact lens or search coil 13

2.3.3 Photo-OculoGraphy (POG) 13

2.3.4 Video-based combined pupil and corneal reflection . . . 13
2.3.5 Eye movement analysis . . . 18
2.4 Mobile eye-tracking hardware . . . 19
2.4.1 Arrington . . . 19
2.4.2 Pupil-pro . . . 20
2.4.3 Tobii . . . 20
2.5 Existing analysis methods . . . 21
2.5.1 Manual analysis . . . 22
2.5.2 Marker-based analysis . . . 23
2.5.3 Semantic analysis . . . 23
2.6 Application domains . . . 25
2.7 Recorded datasets . . . 28
2.8 Challenges . . . 31
2.9 Conclusion . . . 34

3 Object recognition 35

3.1 Introduction . . . 35
3.2 Related work . . . 37
3.3 Approach . . . 42
3.4 Semi-automatic analysis . . . 45
3.5 Results . . . 46
3.6 Conclusion . . . 49

4 Person detection 51

4.1 Introduction . . . 51
4.2 Related work . . . 53
4.3 Approach . . . 58
4.4 Semi-automatic analysis . . . 64

4.5 Person re-identification 64
4.6 Results 65
4.7 Conclusion 70

5 Hand detection **71**
5.1 Introduction 72
5.2 Related work 74
5.3 Semi-automatic analysis 77
5.4 Approaches 78
5.4.1 Model-based approach 78
5.4.2 Segmentation-based approach 88
5.5 Results 95
5.5.1 Datasets 95
5.5.2 Accuracy model-based approach 96
5.5.3 Accuracy segmentation-based approach 98
5.5.4 Computational time 100
5.6 Conclusion 101

6 Gesture detection **103**
6.1 Introduction 103
6.2 Related work 106
6.3 Approach 110
6.3.1 Rest position 111
6.3.2 Gesture segmentation 113
6.3.3 Usage of gesture space 115
6.3.4 Gesture directionality 116
6.4 Results 118
6.4.1 Accuracy of the hand annotations 119

6.4.2 Accuracy of gesture phase segmentation 120
6.4.3 Accuracy of gesture space annotation 122
6.4.4 Output of gesture directionality 123
6.5 Conclusion 124

7 Large scale experiments **127**
7.1 Introduction 128
7.2 Statistical analysis 129
7.3 Analysis of customer journey experiment 133
7.4 Analysis of a triadic conversation 136
7.5 Analysis of lecture recording 140
7.5.1 Body parts versus presentation screen 140
7.5.2 Speaker versus slides 142
7.5.3 Gesture analysis 143
7.6 Visualization of data 146
7.6.1 Visualisation of numerical data 147
7.6.2 Timeline visualisation 148
7.6.3 Heat map visualisation 150
7.7 Conclusion 150

8 Conclusion and Future work **153**
8.1 Conclusion 153
8.2 Future work 155

Bibliography **159**

List of publications **173**

Curriculum Vitae **177**

List of Figures

1.1 Left: illustration of screen-based eye-tracker. Middle: illustration of a mobile eye-tracker consisting of a scene camera and an eye camera. Right: output of a mobile eye-tracker, green dot represents gaze point. 2

1.2 Graphical representation of the workflow of our (semi-)automatic analysis. Hand icons represent the places where manual intervention can be requested. 4

2.1 Anatomy of the human eye. Image from [1]. 10

2.2 Subject wearing electrodes for EOG eye movement experiment. Image from [46]. 13

2.3 Example of a search coil embedded in a contact lens. 14

2.4 Four different Purkinje reflections that are formed when IR light (L) is emitted closely to the eye. Image from [4]. 15

2.5 Relative positions of pupil and first Purkinje images as seen by the eye camera. Image from [46]. 16

2.6 (a) Example of a screen-based eye-tracker(SMI RED500) that is mounted underneath a traditional monitor. Image from [6]. (b) Example of a mobile eye-tracking experiment in a shopping context. Image from [5]. 17

2.7 Example frame of the eye camera. The bright dot below the pupil is the first Purkinje reflection. 17

2.8 Hypothetical eye movement signal. Image from [46]. 19

2.9 From left to right: Arrington [7], Pupil-pro [77] and Tobii [5]. . 19

2.10 Examples of traditional screen-based eye-tracking output. The left part is an example of a heat map output, while in the right part a gaze plot is shown. Both images from [5]. 22

2.11 Illustration of both AOA (green region) and AOI (red region) in a marker(orange squares)-based analysis approach. Image from [3]. 24

2.12 Example frames of various mobile eye-tracking recordings that were made throughout this PhD. 32

3.1 High-level overview of current analysis methods and how our approach fits between them. 36

3.2 Illustration of basic feature matching. Coloured circles represent the features, blue lines represent the matching feature pairs across both images. 38

3.3 The leftmost image includes the object that needs to be retrieved in the other images. 40

3.4 Comparison between ORB and BRISK. The horizontal axis represents the size of the ROI square. The vertical axis represents the amount of detected keypoints. 40

3.5 Illustration of our feature matching, Blue lines illustrate corresponding features. Part(a) represents a valid feature matching, part(b) represents feature matching in which the object of interest is invisible. 44

3.6 (a) Cut-out of our timeline visualisation in which we can distinguish correct and false object detections. (b) Illustration of the feature space of our expanded database. 46

3.7 Precision-recall curve of our object recognition technique tested on a set of 2000 images. Corresponding objects are shown at the right side of the graph. 47

3.8 Improvement of accuracy when additional correct ROIs are added to the image database. 48

4.1 Illustration of a Haar-feature, which is used for face detection. When this feature overlaps with the eye-nose-eye region, it results in a high score due to the fact that the eyes are most often darker than the nose. 55

4.2 (a)Root HOG model, (b) part models representing the limbs and head of a person, (c) The deformation cost for each of the parts with respect to the root model. Image from [51]. 57

4.3 Top-row: computed ICF channels on an image patch. Bottom-row: distribution of the selected rectangular features. Image from [41]. 57

4.4 Example image in which a person is gesturing and difficult to detect using a rigid model-based approach such as HOG. 59

4.5 Three components of the upper body model, each with their own root model(a), part model(b) and deformation model of the parts(c). 60

4.6 Comparing the accuracy of the upper body (UB) detection model against full person detection models on the INRIA test set. . . . 61

4.7 Example of the upper body (component 2 and 3 from our model) and head-shoulder (first component from our model) detections. 62

4.8 Temporal smoothing detection results. Vertical bars: real detections, dashed line: output of the temporal smoothing. . . 64

4.9 Sample frame of human-human experiment with three participants. Image from the scene camera of the third participant. . 66

4.10 Histogram comparison used for person re-identification. 66

4.11 Precision-Recall curves of our upper body detection implementation compared to a standard model. 68

4.12 Precision-Recall curves of face detection compared to upper body detections. 70

5.1 Illustration of human-human interaction. The red dot represents the current visual focus of the subject wearing the mobile eye-tracker. 73

5.2 Graphical representation of the model-based hand detection approach. The three stages: upper body and face detection, hand detection and a combination of elimination and tracking. 78

5.3 Illustration of the hand model(a), and an illustration of some sample images that were used for training the hand and context models(b). 80

5.4 Illustration of the rotation of our images in order to detect hands in any orientation. Left: step size is 10° per rotation. Right: step size is 20° per rotation. 82

5.5 From left to right: original image(a); binary image based on skin segmentation(b); skeletonization(c); arm and hand estimation(d). 83

5.6 Left: large amount of detections before elimination; Right: final clusters after elimination step. 84

5.7 Illustration of our reduced orientation concept. Top part: hypothesis of orientations that would result in the best detection scores. Bottom part: actual rotations that obtained the best detection scores. 86

5.8 Accuracy of hand model that is applied on a limited number of rotated images. Blue curve represents the accuracy of all orientations. Other curves represent the reduced orientations. . 87

5.9 Workflow of our segmentation-based hand detection approach. 89

5.10 Generation of hand candidates: a) original image, b) skin segmentation, c) contour detection, d) fit ellipse, e) final hand candidates. 90

5.11 Examples of our detections on the four datasets. Green circles are the hand detections, yellow circles are the corresponding joints, red circles indicate the estimated shoulder positions. 91

5.12 Example frames from the Buffy dataset [52] indicating the large variety of human poses within this set. From this labelled dataset, our probability maps (P_{Elbow}) and (P_{Wrist}) are derived. 91

5.13 Top image shows data points and probability map of the left wrist w.r.t. left shoulder(P_{Wrist}). Bottom image shows the data points and probability map of left elbow w.r.t. left shoulder(P_{Elbow}). The red dot in each map illustrates the position of the left shoulder. 93

5.14 Examples in which the shoulder-joint position was wrong. (a) left arm: joint and hand are swapped (b) left arm: both joint and hand are wrong (c) right arm: position of joint is completely wrong. 94

5.15 Illustration of each of the datasets used for the validation of our hand detection approach. 96

5.16 Result of our (semi-)automatic approach in which accuracy is improved by manual interventions. 99

5.17 Result of hand model applied on images where manual intervention was requested. 101

6.1 (a)Example frame of NeuroPeirce corpus [20]. (b) Example frame of SaGA corpus [88]. 106

6.2 Workflow of our automatic gesture analysis tool. This figure also reveals which type of analysis results in XML compatible output. 111

6.3 Normalised hand positions of an entire recording. Green dots represent right hands, red dots represent left hands. Two asterisks indicate the respective rest positions. 112

6.4 Top part: displacement of the right hand w.r.t. the rest position. Red line indicates the applied threshold. Bottom part: gesture segmentation that is generated using this displacement. 114

6.5 Gesture space as defined in [93]. We can distinguish 4 larger sectors as represented by capital letters as well as the respective sub-sectors. 116

6.6 Automatically generated gesture space based on the upper body and face detections. Here the left hand is located in the *periphery*, while the right hand is located in the *extreme periphery*. Image obtained from the NeuroPeirce corpus [20]. 117

6.7 Validation methodology that is used for the gesture phase segmentation. 120

6.8 Precision-recall curves of our approach for both recordings. Coloured circles represent the accuracy of the AUVIS [116] method. 121

6.9 Examples of gestures captured into a single image. 124

7.1 Screenshot of the ELAN annotation software in which a gaze and gesture tier of an eye-tracking recording are shown. 130

7.2 Examples of selected objects of interest in the context of the museum experiment. 135

7.3 Results of our algorithm applied to the recordings of the museum visit. Each timeline represents a short summary of viewing behaviour of a participant. 137

7.4 Different objects and persons that were automatically labelled using our software. 138

7.5 Example in which there is disagreement between manual and automatic annotation. The gaze cursor is indeed positioned between the speaker and the camera tripod. 139

7.6 Selected objects of interest for the analysis of the lecture recording. 144

7.7 Variations in upper body detections that may cause changes in relative hand positions. 146

7.8 Illustration of a participant who is looking at a gesture that is made by the speaker. 147

7.9 Visualisation methods for representing the numerical analysis data of an eye-tracking experiment. 149

7.10 Automatically generated timelines of an entire experiment. . . . 151

7.11 Heat map of an eye-tracker experiment. 152

List of Tables

3.1 Experimental comparison of local feature extraction methods. . 39

3.2 Computational time of the object recognition implementation. . 49

4.1 Computational cost of our upper body model tested under various parameter settings. 69

5.1 Accuracy of the hand model versus rotation angle of the images. 80

5.2 F_1-measure of our model-based approach both with and without tracking and compared against other hand detection approaches. In this experiment the option for manual interventions was disabled. 97

5.3 Comparison of our segmentation-based hand detection approach and our model-based approach. *man.* indicates the amount of hands that were manually annotated. 99

5.4 Execution times per frame averaged over all frames of 1280×720 pixels. 101

6.1 Measurements of the semi-automatic hand annotation of both recordings. 119

6.2 Accuracy of the optimal working point i.e. $\alpha = 0.9$. Bottom row of the table shows the best accuracy of the AUVIS tool tested on the same corpora. 122

7.1 Questions to be answered in the context of the museum visit. . 135

7.2 Reliability of triadic analysis. 139

7.3 Reliability of lecture analysis. Items under scrutiny: face, upper body, hands and presentation screen. 142

7.4 Reliability of lecture analysis. Items under scrutiny: speaker and each individual slide. 143

7.5 Reliability of gesture analysis. 145

Chapter 1

Introduction

Human eye movements have attracted scholarly attention for a long time. Well-known research domains with a long-standing interest in gaze behaviour are psychology, market research, human-human interaction, sports and kinematics, linguistics, etc. See [45, 62, 109, 131, 137] for an overview. Well-known applications include research on reading, scene perception and visual search tasks. In general, eye movement research involves the determination of *where a person is looking at*. In order to gain fine-grained information on the distribution of visual attention, eye-tracking systems are used. Such systems typically consist of a monitor to which one or multiple cameras and IR-illuminators are attached. These are pointed towards the person in front of the monitor and capture the eye movements. By combining the eye positions (gaze locations) and the content on the monitor, one is able to investigate the visual behaviour. Such a methodology enables research on scene perception during visual search tasks, decision-making, etc. Such screen-based eye-trackers allow for valuable experiments since it is straightforward to iterate the same experiment over multiple participants by using the exact same stimuli.

In the last four decades, eye-tracking research has established itself as a powerful paradigm for studying human visual behaviour. In a more recent development, efforts have been made to extend the field of application for eye-tracking research beyond the boundaries of the laboratory. For example research on marketing or on human-human interaction could definitely benefit from such real-life experiments. In 1999, Michael Land [85] was one of the pioneers using head-mounted eye-trackers. i.e. wearable devices designed for capturing eye movements in a natural setting. The development of these mobile eye-tracking systems has opened up the paradigm of eye-tracking to a wide variety of research

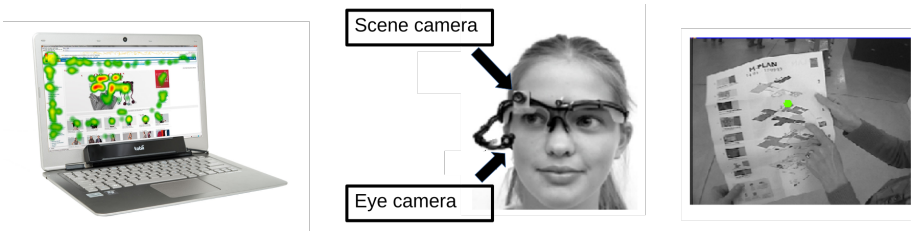


Figure 1.1: Left: illustration of screen-based eye-tracker. Middle: illustration of a mobile eye-tracker consisting of a scene camera and an eye camera. Right: output of a mobile eye-tracker, green dot represents gaze point.

disciplines and commercial applications. Whereas traditionally, the analysis of eye gaze patterns was largely confined to controlled lab-based conditions due to technological restrictions (i.e. obtrusive hardware restricting the flexibility of use and potential research questions), mobile systems allow for eye-tracking 'in the wild', without a necessarily predefined set of research conditions. Because of this increased flexibility, research on visual behaviour and real-life user experience now extends to natural environments such as public spaces (train stations, airports, museums, etc.), commercial environments (supermarkets, shopping centres, etc.) or to interpersonal communicative settings (helpdesk interactions, lectures, face-to-face communication, etc.).

A mobile eye-tracker, as illustrated in figure 1.1, combines two types of cameras. The scene camera is looking forward and captures the field of view, while the eye camera(s) on the other hand capture the eye movements, known as gaze data. Output of such an eye-tracker, as shown in the right part of this figure, consists of the images captured by the scene camera on which the gaze data is superimposed.

In 2011, Hayhoe et al. [61] presented a clear comparison between mobile and traditional lab-based eye-tracking. A first difference is found in scene perception. Visual input in the real world is three-dimensional (3D) and varies constantly as a consequence of the observer's movements in the scene, whereas in screen-based eye-tracking the stimuli are not manipulated by the observer and interaction is limited. Another difference can be found in the underlying task one is performing. In screen-based eye-tracking, one is most likely occupied with recognising and remembering the objects in the scene. In real-life eye-tracking on the other hand, subjects need specific information for navigation, obstacle avoidance, etc. It is clear that real-life experiments are crucial in research on understanding the principles that control gaze and the selective acquisition of visual information from scenes.

Although both screen-based and mobile eye-tracking tackle the same underlying objective, i.e. gaining insights into the visual behaviour of a participant, a crucial difference exists between the analysis of the recorded data of both approaches. A screen-based eye-tracker captures the eye positions of a subject looking at a screen. Analysing such an experiment involves the mapping of the eye gaze on the stimuli. The output of such an analysis includes heat maps, indicating which part of the stimuli received the most visual attention, trajectories of the gaze data, indicating the positions where the gaze cursor halted (fixations) and the trajectories between them (saccades). The screen-based eye-tracker allows for a relatively straightforward analysis since a) the region in which the gaze data is captured is restricted, i.e. the monitor, and b) the content of the stimuli is known at each moment in time. In mobile eye-tracking on the other hand, the analysis is much more complicated.

By abandoning the traditional well-controlled lab-based conditions, the datastream generated by the mobile eye-trackers becomes highly complex, both in terms of the objects and scenes that are encountered, and the gaze data that needs to be analysed and interpreted. How can researchers avoid the painstaking task of manually coding large amounts of data, which is extremely time-consuming, without losing the full potential of mobile eye-tracking systems? Available solutions exist, however they often require experimental control, making real-life experiments practically infeasible. On top of that, most solutions are not applicable in each type of experiment like for example human-human interaction recordings. Researchers in this field are interested in the dynamics of how and when conversational partners establish, maintain and break down joint gaze (i.e. jointly focusing on a specific object of interest) and mutual gaze (i.e. eye contact).

In order to overcome the challenging analysis of mobile eye-tracking data, we initiated a multidisciplinary research project i.e. InSightOut[1] in 2012. Here, our goal was to build a bridge between computer vision research on one hand and linguistics on the other hand. Two research groups were involved in this project: EAVISE (Embedded Artificially intelligent VISion Engineering) under supervision of Prof. dr. ir. Toon Goedemé, and MIDI (Multimodality, Interaction & Discourse) under supervision of Prof. dr. Geert Brône. The goal of this project was to investigate whether and how computer vision algorithms could facilitate the analysis of mobile eye-tracking recordings, for example we investigated the use of object recognition algorithms to automatically determine which object or item the subject is looking at. This collaboration formed the basis of this PhD research in which the integration of computer vision techniques for the analysis of mobile eye-tracker data was further explored.

[1] http://www.eavise.be/insightout/

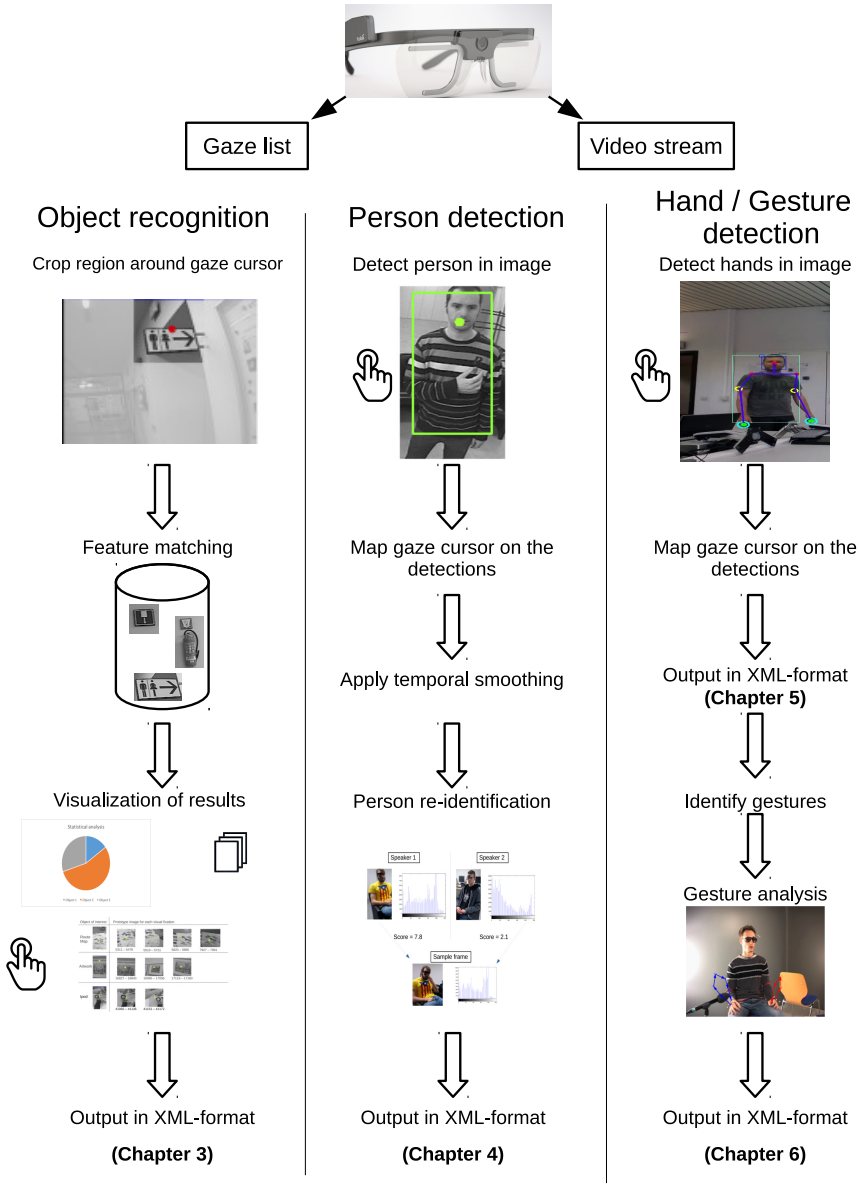


Figure 1.2: Graphical representation of the workflow of our (semi-)automatic analysis. Hand icons represent the places where manual intervention can be requested.

In figure 1.2 a graphical overview of our generic framework for the analysis of mobile eye-tracking recordings is given. It is clear that we focus on three main aspects. (i) Object recognition, which is used to automatically determine which object or item the subject is looking at. (ii) Person detection, allowing for the automatic determination of how long and how often a specific person is the focus of visual attention. (iii) Hand and gesture detection, which is used to automatically determine whether the visual behaviour is influenced by particular gestures.

Next to these three main aspects, we also propose a novel integration of manual intervention within our automatic analysis. Such an approach is unique since in general, computer vision algorithms are used to make human input superfluous. Due to the challenges in terms of the objects and scenes that are encountered in mobile eye-tracking experiments (e.g. rapidly moving camera position as well as moving objects in the scene), we conclude that a fully automatic approach will not reach the required accuracy. To ensure an analysis that is as accurate as possible, we allow for manual intervention in order to steer the automatic analysis. In figure 1.2 the integration of this manual intervention step is indicated using the *hand* icon. In each aspect our framework, the manual analysis is intertwined.

Next to the analysis of mobile eye-tracking experiments we also paid attention to the usability of our framework. To increase the applicability, we foresee an interface for a range of output formats including:

- Statistical data, which are mainly used for marketing applications.
- A timeline representation of an entire experiment, which is of vital importance in for example customer journey experiments.
- A final output format that is an XML-based file making our approach integratable with commonly used annotation tools such as ELAN or ANVIL.

It is important to recapitulate that our goal is to speed up the processing of mobile eye-tracking data with several orders of magnitude while maintaining high accuracy. This is done using the integration of computer vision algorithms that are combined with the ability to perform manual interventions. Using these interventions, we give users a certain level of control over the automatic annotation process.

In the remainder of this chapter we give a summary of the main contributions of our work in section 1.1, and a comprehensive overview of the outline of this dissertation in section 1.2.

1.1 Main contributions

The main contributions that we propose throughout the different chapters in this dissertation can be summarised as follows:

- We introduced computer vision techniques in the field of mobile eye-tracking. To exploit the full potential of mobile eye-trackers the majority of existing studies are based on the manual annotation of the generated data. Using our semi-automatic analysis framework, we can perform the analysis far more efficient, making the processing of larger datasets possible.
- We propose a generic framework for the analysis of mobile eye-tracking recordings including object, person, face, hand and gesture detection algorithms. The term *generic* was consciously chosen since our approach can be applied for other purposes as well, which is further discussed throughout this dissertation.
- Since the goal is to analyse these mobile eye-tracking experiments as accurately as possible, we propose the novel integration of manual interventions in an automatic analysis system. This combination yields an extremely high accuracy at a minimum cost of manual interventions. Again, such an approach is not limited to this application, but the same methodology can be used in other applications as well.
- This PhD involves a multidisciplinary set-up in which computer vision and linguistics are brought together. Throughout the entire trajectory of this PhD there was a close and intense collaboration between myself as member of the EAVISE research group and the MIDI research group, which has built up experience in both mobile eye-tracking and the analysis of the obtained data over the last decade. I believe it is fair to state that both research groups learned a lot from each other.
- We recorded several hours of mobile eye-tracking data in a variety of settings. Amongst these recordings are experiments of customer journeys in Museum M (Leuven), presentations, wayfinding, human-human interaction, rehearsals of musicians, etc. Many of these recordings are labelled in terms of relevant objects or items and some of them were made publicly available for other researchers. By actively participating in each of these recordings we became experienced in various aspects of mobile eye-tracking.

1.2 Outline of this dissertation

To situate and motivate this research topic, we first give an overview of the developments in (mobile) eye-tracking in **chapter 2**. We discuss the current approaches on the analysis of mobile eye-tracking data and discuss their advantages and disadvantages. Next we motivate why a computer vision-based analysis should be developed and we define which specifications such an approach should achieve in order to compete with the current methods. Furthermore, we discuss some challenges that are involved using low-cost eye-trackers. Especially when multiple of these low-cost devices are used simultaneously, problems arise.

In **chapter 3** we present our object recognition approach. First we give an overview of existing approaches to the recognition of specific objects. Next we propose our approach in which recent techniques are combined to achieve high recognition rates at a minimal computational cost. We also propose the integration of manual intervention in our object recognition approach and we highlight the benefits of this integration. Finally, we present accuracy measurements that were generated by the automatic analysis of a wayfinding experiment.

We propose a human person detection approach in **chapter 4**. We start this chapter by discussing state-of-the-art person detection approaches, each with their advantages and disadvantages. Next we explain our approach, which consists of the development of a new person detection model, the combination of both face and human upper body detection as well as several methods to further increase the accuracy of our analysis. Next we discuss the integration of manual intervention, which is used to further boost the accuracy. Furthermore, we introduce a person-re-identification approach which is of great importance in the analysis of complex human-human experiments. Finally, we present accuracy measurements of our person detection approach applied to the recordings of a customer journey experiment in a museum.

In **chapter 5** we start by introducing applications of mobile eye-tracking that could benefit from an automatic detection of human hands. We also give a thorough overview of existing hand detection approaches. Next, we propose the integration of manual input in our automatic hand detection algorithm since achieving top accuracy is of vital importance in these applications. In this chapter, we propose two approaches for the detection of human hands in images. Finally, we compare both approaches both in terms of accuracy and computational cost. These validation experiments were performed on recordings of human-human interaction experiments.

Our gesture detection algorithm is proposed in **chapter 6**. We start by presenting existing approaches for the detection and segmentation of gestures

and we clearly explain why we developed our own approach. Besides that, we also explain how such a gesture analysis could contribute to the analysis of mobile eye-tracking data. Then we explain our approach which consists of several steps including the analysis of the gesture directionality and the usage of the gesture space. Furthermore, we present a large scale validation of our gesture analysis on two existing corpora. This validation also reveals the generic aspect of our approach since these corpora are not recorded using a mobile eye-tracker. Of course, besides this validation, we highlight the usefulness of our approach in the analysis of mobile eye-tracking data.

We evaluate our entire framework in **chapter 7**. Here we analysed three real-life mobile eye-tracking experiments. The first experiment was conducted as part of a customer journey experiment in a museum context. The second experiment is a recording of a challenging human-human interaction experiment. The analysis of this recording was done using our object recognition, person detection and re-identification. We compare our automatic analysis to manual annotations for both accuracy and efficiency. The latter experiment is a recording of a person attending a presentation. Here we used our entire framework to analyse his visual behaviour both in terms of looking at the speaker and looking at the presentation screen. Furthermore, we automatically analyse the gestures of the speaker and show that our approach can provide some insights into the influence of his gestures on the visual behaviour of our participant. Again, we compare our automatically generated analysis to manual annotations. Finally, we discuss the quantity of manual interventions in these experiments, which reveals that a minimum amount of manual interventions can significantly improve the accuracy of our analysis.

Finally, in **chapter 8** we conclude this dissertation with a summarisation of our work and we give an overview of possible further improvements.

Chapter 2

(Mobile) eye-tracking

Because this PhD thesis is about the automatic processing of mobile eye-tracking recordings, we start by giving the reader a thorough introduction in eye-tracking and mobile eye-tracking in particular. First, we give a short introduction into the anatomy of the human eye and on eye movements. Next, we give an overview of the developments that were made within the field of eye-tracking. Then we give an overview of existing mobile eye-trackers as well as current methods for the analysis of this type of data. Furthermore, we give an outline of applications in which mobile eye-tracking is used. We also present the recordings that we have made during this PhD project and that were used for the validation of our algorithms. Finally, we give an overview of challenges that we need to overcome when developing a framework for the accurate and efficient analysis of mobile eye-tracking data.

2.1 Anatomy of the eye

In this initial section, we give a brief overview of the anatomy of the human eye and clarify several terms that will be used in this thesis. In figure 2.1 a graphical representation of a human eye is given. A short introduction of relevant terms:

- Lens: focusses light rays to the retina
- Retina: sensory membrane that receives images formed by the lens and converts them into signals to the brains

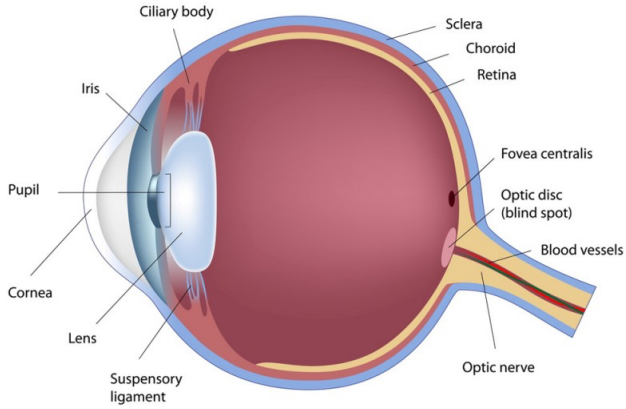


Figure 2.1: Anatomy of the human eye. Image from [1].

- Pupil: the round dark circle of the eye that opens and closes to regulate the amount of light the retina receives
- Iris: the coloured part of the eye that surrounds the pupil. The iris acts like a diaphragm, and hereby opens and closes the pupil
- Cornea: the clear part of the eye that covers the iris and the pupil
- Sclera: the white of the eye
- Limbus: the border between cornea and sclera
- Fovea: the part of the retina that is responsible for accurate sight

The eyes move within six degrees of freedom: three rotations and three translations within the socket. Six muscles are responsible for the movement of the eyeball: the medial and lateral recti (sideways movements), the superior and inferior recti (up/down movements), and the superior and inferior obliques (twist) [33].

2.2 Eye movements

Eye-tracking is a technique that involves the recording of the movements of the human eye. In order to gain some first insights into the notion of eye movements, we present a basic overview of relevant terminology. In [46] a more profound

overview of the taxonomy and modelling of eye movements is given, of which we give a brief summary below.

In eye movements a distinction can be made between movements used to reposition the fovea and other movements such as adaptation and accommodation, which refer to non-positional eye movements (e.g. pupil dilation or lens focusing). With respect to eye-tracking, our focus lies on the first type of movements. In general, four types of eye movements can be distinguished: saccades, fixations, smooth pursuits and nystagmus.

Saccades

Saccades refer to rapid eye movements that are used to reposition the fovea to a new location in the visual environment. Saccadic movements can be voluntarily executed or they can be reflexive. The duration of a saccade lies in the range of 10 ms up to 100 ms, which make them fast enough to be unnoticeable for humans [121].

Fixations

Fixations are the eye movements that maintain the visual gaze on a single location. Fixations are characterised by miniature eye movements such as micro saccades, drift and tremor. The minimum duration of a fixation varies from 150 ms up to 600 ms and one could assume that 90% of viewing time is devoted to fixations [63]. Indeed, the fixations are the eye movements that provide information of visual attention and are therefore the most important ones concerning eye-tracking.

Smooth pursuits

Pursuit movements occur when visually tracking a moving target. For example: to make a smooth pursuit movement: look at your thumb, at an arms length and move your arm from left to right while fixating your fingertip. The eyes are capable of matching the velocity of a moving target depending on the range of target motion. A smooth pursuit is a perfect example of a closed-loop feedback loop [26]. Here, the signals from the visual receptors introduce an error signal, which indicates the needed compensation to match the motion of the moving target.

Nystagmus

Nystagmus eye movements are typically characterised by a sawtooth-like time course. A well-known example of Nystagmus occurs when one looks at a stationary object from a moving train. The eyes will follow the object slowly (smooth pursuit), but will then rapidly jump back (saccade), whereafter the process will repeat itself.

To summarise this section, we can conclude that three types of eye movements are relevant to gain insights on visual attention: saccades, fixations and smooth pursuits, since they are all in one way or another responsible for visually fixating on specific items within the field of view. These items can be either stationary (fixation) or moving (smooth pursuits). The saccades on the other hand make the eye jump from one visual fixation to the other. More information regarding eye movements can be found in [26].

2.3 Eye-tracking techniques

The device for measuring eye movements is commonly known as an eye-tracker. In chapter 1, we already introduced two eye-tracking techniques, viz. screen-based eye-tracking and mobile eye-tracking, which are the most recent techniques. In this section we discuss the history of eye-tracking. We can distinguish four broad categories: Electro-OculoGraphy (EOG), scleral contact lens/search coil, Photo-OculoGraphy (POG) and video-based combined pupil and corneal reflection. We refer the reader to [46, 111] for more information on eye-tracking methodology.

2.3.1 Electro-OculoGraphy (EOG)

During the mid-70's, EOG was the most applied method for eye movement research [141]. Today, this technique still exists, however it is rarely used. The EOG technique relies on measuring the electric potential differences of the skin using electrodes placed around the eye. In figure 2.2, an illustration of a subject wearing the electrodes is shown. The recorded potentials measure around 15-200 μV, whereas the sensitivity is around 20 μV per degree of eye movement. This approach measures the eye movements relative to the head position, making it unsuitable for most point of regard (POR) experiments. However, if the head position is tracked as well or if the head is fixated with an external apparatus, a wider range of experiments could be explored.

Figure 2.2: Subject wearing electrodes for EOG eye movement experiment. Image from [46].

2.3.2 Scleral contact lens or search coil

A scleral contact lens approach measures the eye movements using a mechanical or optical reference object, which is mounted directly on the eye using a contact lens. Such a lens is significantly larger than traditional lenses since slippage of the lens should be avoided. Therefore, such a lens covers both cornea and sclera. A commonly used method implies the attachment of a wire coil to the contact lens as illustrated in figure 2.3. This coil is then measured moving through an electromagnetic field. Experiments have shown that the scleral contact lens is the most accurate method for eye movement research [141]. However, it is also the most intrusive method. Inserting the lens is challenging and requires practice, and wearing the lens causes discomfort. Another disadvantage is that this method measures eye movements relative to the head position.

2.3.3 Photo-OculoGraphy (POG)

The POG or Video-OculoGraphy (VOG) entails a variety of techniques for eye movements recording involving the measurement of distinguishable features. Examples are the apparent shape of the pupil, the position of limbus and the corneal reflections of a direct light source, which is often a closely situated infra-red (IR) source. Most of these methods require a fixed head position, which is often achieved using a head or chin rest or a bite bar [141].

2.3.4 Video-based combined pupil and corneal reflection

As mentioned above, the previous techniques generally are not suitable for POR experiments. To provide these measurements, either the head must be fixed,

Figure 2.3: Example of a search coil embedded in a contact lens.

or multiple ocular features must be extracted in order to disambiguate eye movement from head movement. An example of the latter one is the extraction of both corneal reflection and the pupil centre, which is used in video-based eye-tracking. Here both an IR illumination and camera are pointed towards the eye, so that the angle of the eye can be estimated by measuring the relative position of the reflection of the IR illumination.

In video-based eye-tracking, two main techniques can be distinguished: screen-based eye-tracking (although sometimes this technique is used without a screen, as in for example the table-top experiments of Jokinen et al [69]) and mobile eye-tracking (also known as head-mounted eye-tracking). Despite their differences, the optics of both systems are highly similar and include relatively inexpensive cameras and image processing techniques to compute the POR in real-time.

When light is emitted to the eye, four different corneal or Purkinje reflections are formed due to the construction of the eye [31], as shown in figure 2.4.

- P1: reflection from front surface of the cornea
- P2: reflection from rear surface of the cornea
- P3: reflection from front surface of the lens
- P4: reflection from rear surface of the lens

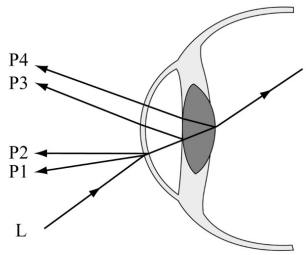


Figure 2.4: Four different Purkinje reflections that are formed when IR light (L) is emitted closely to the eye. Image from [4].

Typically, the eye camera of a video-based eye-tracker locates the first Purkinje reflection. However, some dual Purkinje eye-trackers exist, in which both first and fourth reflections are tracked.

Besides the corneal reflection, another reference point is needed to separate eye movements from head movements. Therefore, the pupil centre is detected as well. Two methods exist: dark and bright pupil tracking. In bright pupil tracking the IR illuminator is placed close to the optical axis of the eye camera, causing the pupil to appear brighter than the surrounding areas. In dark pupil tracking, the IR illuminator is placed away from the optical axis, making the pupil appear darker than the iris.

The relative position between pupil centre and the first Purkinje reflection changes due to eye movements, however it remains relatively constant with minor head movements. In figure 2.5 the relative positions between the first Purkinje reflection and the pupil are shown. Here we see the eye fixating at nine calibration points. The Purkinje reflection is indicated by the white circle, while the pupil is illustrated by the black circle. Since the IR light is typically placed at a fixed position relative to the eye, the Purkinje reflection is relatively stable. The pupil, on the other hand, moves in its orbit. Based on the displacement between both reference points, one is able to identify the point of regard.

As mentioned before, video-based eye-tracking consists of two main approaches: screen-based and mobile eye-trackers. Both approaches are discussed below.

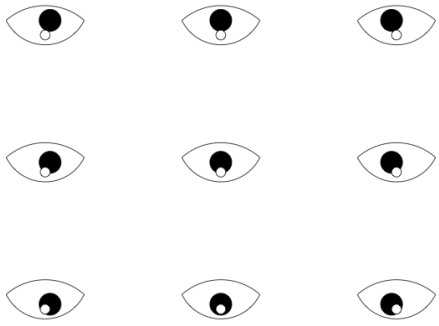


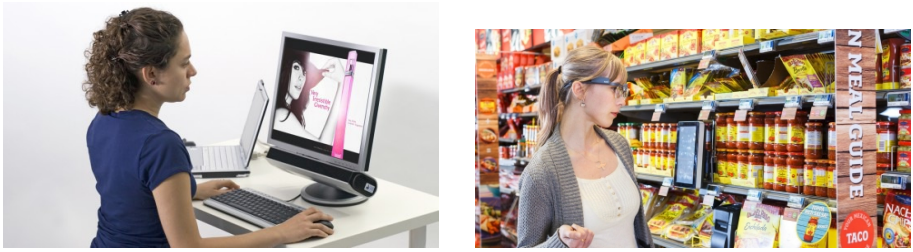
Figure 2.5: Relative positions of pupil and first Purkinje images as seen by the eye camera. Image from [46].

Screen-based eye-trackers

Typically, a screen-based eye-tracker consists of a traditional flat panel display, in which a camera and IR illuminators (often LEDs) are embedded underneath the screen as shown in figure 2.6(a). This setup is completely unobtrusive making it highly applicable. Calibrating such a setup is commonly accomplished by making the subject look at a number of predefined positions at the screen, while simultaneously recording both Purkinje reflection and pupil centre. Based on these calibration points (normally nine or more, however recent systems use a one-point calibration as well), one is able to map the position of the eye to the respective position on the screen. Screen-based eye-tracking allows for a rather limited freedom of movement, i.e. the head may move within a zone of approximately 50×40 cm. There is a wide variety of applications in which screen-based eye-tracking is used. Examples are psychology and neuroscience in which reading research is performed [8] and, user experience and marketing research in which one evaluates websites, media and commercials, etc.

Mobile eye-trackers

The second type of video-based eye-tracking devices are the mobile eye-trackers also known as head-mounted eye-trackers. These devices are wearable, thus the entire eye-tracking infrastructure is mounted onto the head of a subject. Here, one or multiple cameras and IR illuminators are mounted onto a wearable frame, which is highly similar to a pair of glasses, and are pointed towards the eyes. An example of a frame captured by the eye camera is given in figure 2.7. Since the device is wearable, it allows for eye-tracking recordings outside lab-conditions. These real-life recordings enable insights into visual behaviour in



(a) (b)

Figure 2.6: (a) Example of a screen-based eye-tracker(SMI RED500) that is mounted underneath a traditional monitor. Image from [6]. (b) Example of a mobile eye-tracking experiment in a shopping context. Image from [5].

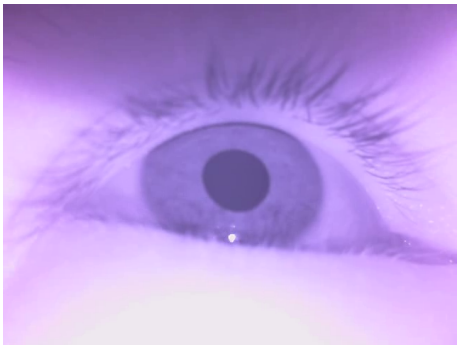


Figure 2.7: Example frame of the eye camera. The bright dot below the pupil is the first Purkinje reflection.

natural and unrestricted environments such as supermarkets, public buildings, etc. An illustration of such an application is given in figure 2.6(b) in which a mobile eye-tracker is used in a real-life shopping experiment. Furthermore, an additional camera is pointing forwards and captures the field of view of the subject. Again, a calibration step is required before one can use the eye-tracker. This calibration is similar to the one described above, however instead of presenting the calibration point on the screen, here the points are presented in the real world. More recent mobile eye-trackers are able to perform the calibration using a single point.

Although mobile eye-trackers allow for a broader range of experiments, some criticism exists on the concept of mobile eye-tracking. It seems that some researchers are sceptic to use them in their research, just because there is so much flexibility. They claim that it is difficult or even impossible to repeat the

same experiment under the exact same conditions, making these experiments methodologically problematic. On top of that, two key assumptions that were presented by Just and Carpenter [72], and that link cognitive processing and eye movements may be particularly problematic in the context of mobile eye-tracking. First, the *eye-mind assumption* states that words or objects are fixated as long as they are being processed. This means that there is a close relationship between what the eyes are gazing at and what the mind is engaged with. In real-life eye-tracking experiments, however, many fixations are caused/guided by subconscious processes that have to do with perception-action coupling rather than cognition. The second assumption, the *immediacy assumption*, states that words or visual objects that are fixated by the eyes are immediately processed. Again, in real-life experiments, we often notice that one is looking at a specific object, while thinking about something completely different (e.g., background noises that are typically absent from experimental conditions). Note that it is the frequent violation of the eye-mind and immediacy assumptions, which is responsible for a well-known problem in gaze-based human-computer interaction: The *Midas Touch Problem* [66]. Because not all fixations are made intentionally to acquire and immediately process the fixated information, it is difficult to disambiguate fixations that were made on purpose (to control the computer) and fixations that occurred unintentionally and therefore result in unwanted actions by the gaze-controlled computer.

In this dissertation, we start from the assumption made in different fields that it is worthwhile to explore gaze behaviour in naturalistic settings, either alone or in combination with other measuring techniques. The applications of mobile eye-tracking are thoroughly discussed in section 2.6.

2.3.5 Eye movement analysis

The data obtained from any eye movement experiment may appear informative, however without further analysis, the raw data is meaningless. Despite one can estimate to what the subject paid attention by looking at the raw data, it is crucial to identify the fixations to indicate the locations of the viewer's visual attention.

A first step is extracting both saccades and fixations, which is done using analysis of the movement signal as illustrated in figure 2.8. Here, the hypothetical plot of an eye movement in time is shown. The analysis task implies that one locates abrupt changes, which indicate the end of a fixation and the start of a saccade. A stationary characteristic, on the other hand, indicates the start of a fixation and the end of a saccade.

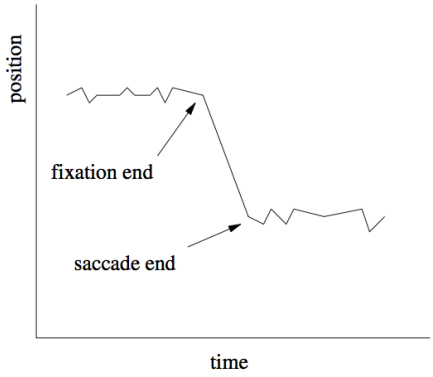


Figure 2.8: Hypothetical eye movement signal. Image from [46].

a) b) c)

Figure 2.9: From left to right: Arrington [7], Pupil-pro [77] and Tobii [5].

2.4 Mobile eye-tracking hardware

Although screen-based eye-tracking yields a lot of experimental possibilities, we focus on a mobile eye-tracking data in this PhD. Indeed, mobile eye-tracking enables more natural experiments, and provides more challenges related to image processing compared to screen-based eye-tracking. During this PhD project, we performed numerous mobile eye-tracking experiments (see section 2.7 for a detailed overview). In this section we give an overview of the equipment we used in our experiments.

2.4.1 Arrington

From the beginning of this PhD study, two mobile eye-trackers were available in the MIDI research group viz. two Arrington Gig-E60 mobile eye-trackers. See figure 2.9(a) for an illustration. This eye-tracker consists of one scene camera which captures a field of view (FOV) of 56° and which records grey-

scale images at a resolution of 320×240 pixels at a frame rate of 24 frames per second (fps). The eye-tracker is monocular, only one eye camera and IR illumination is available. The eye camera records images at 30 fps. The eye-trackers are easily configurable, i.e. both the position of the eye camera and IR illumination is adaptable, making it easy to obtain a clear view of the eye of any participant. Although these are relatively old devices (they were bought in 2009), they provide highly accurate measurements. This system does have a few disadvantages: the resolution of the scene camera is rather limited and on top of that it only captures grayscale images. Furthermore, they are not highly mobile (i.e. a large and heavy battery pack and laptop are required during the recording) which can make recording out of lab conditions challenging.

2.4.2 Pupil-pro

Throughout this PhD project, we acquired new mobile eye-trackers since a) the quality of the Arrington scene camera was inadequate and b) for some experiments three mobile eye-tracker were required simultaneously. During our quest for new equipment, we discovered a new brand of mobile eye-trackers: Pupil, which developed an open source low-cost mobile eye-tracking platform [77], see figure 2(b). The frame of their eye-tracker is built using 3D printing. The scene camera is a standard USB webcam of which the lens has a 90° diagonal FOV and which records colour images at 30 fps of maximum 1920×1080 pixels. The eye camera on the other hand captures images of 800×600 pixels at 30 fps. Furthermore, they developed a software framework for eye detection as well as a graphical user interface to playback and visualise the gaze data. The purchase price of their system is €1390, which is cheap for this type of equipment. However, this low price comes at a cost. Since both cameras are connected using an USB interface, their frame rate depends on the load of the operation system of the connected computer. Therefore, it might happen that sometimes frames are dropped in the recordings. During our experiments we also noticed that it is sometimes challenging to perform an accurate calibration. Given both advantages and disadvantages, we can conclude that these eye-trackers are applicable for basic experiments, however when high accuracy is required, another brand is preferable.

2.4.3 Tobii

A brand that should not be missed in an overview of (mobile) eye-tracking vendors is Tobii. Their latest mobile eye-tracker version, the Tobii Pro Glasses 2, is a state-of-the-art mobile eye-tracker. As shown in figure 2.9(c), it looks highly

similar to a normal pair of glasses, which increases its potential use in real-life experiments. Their system embeds 4 eye cameras, which sample the gaze data at 50 Hz. The 2 cameras per eye make their system insensitive to displacements of the eye-tracker. The scene camera records images of 1920×1080 pixels at 25 fps. On top of that, their system stores the entire recording internally, making it unnecessary to carry a laptop or other recording device during the experiment. However, these advanced features come at a cost of approximately €20.000 per mobile eye-tracker and software, making them very expensive.

Although there are other mobile eye-tracker vendors (SMI, Ergoneers, SensoMotoric,ASL, etc.), the above mentioned survey gives a clear view on the existing equipment in both top and lower segment.

2.5 Existing analysis methods

In the previous section, we gave an overview of several mobile eye-tracking devices. In this section, we discuss existing methods for the analysis of the recorded data. Eye-tracking experiments are mostly performed in order to measure how often and for how long the test subjects looked at a specific object and/or at persons, to gather information about what 'catches the eye' in a certain setting. As mentioned before, the analysis of screen-based eye-tracking experiments is straightforward since the content the subject is looking at is known in advance and is highly controllable. By simply mapping the fixations and saccades on the corresponding positions on the screen, one gets insight in the visual attention of the subject. Output of a screen-based eye-tracking experiment is often rendered as heat maps or gaze plots as shown in figure 2.10. The red regions on the heat map indicate locations on the screen that attracted the most visual attention. The circles on the gaze plot indicate the locations of the fixations, numbered in time order. With these visualisations, researchers can quickly get an idea of which positions on the screen are looked at the most and in which order. This of course only makes sense for relatively static screen content, such as a website or a publicity poster.

In case of mobile eye-tracking on the other hand, the analysis is far more complex since the visual input is mostly unknown in advance and differs between experiments. Compared to screen-based eye-tracking, there is no fixed reference frame in which the analysis is done. Recently, several solutions to the analysis problem have been proposed, some of which have been integrated in commercially available systems. See [49] for an overview.

In the next subsections, some of these existing analysis methods are described, starting with manual analysis.

(a) (b)

Figure 2.10: Examples of traditional screen-based eye-tracking output. The left part is an example of a heat map output, while in the right part a gaze plot is shown. Both images from [5].

2.5.1 Manual analysis

The oldest method for analysing mobile eye-tracking data is manually analysing each individual frame of the scene camera on which the corresponding gaze data is superimposed. Depending on the used infrastructure, it is possible to restrict the analysis to the detected fixations, however these are responsible for 90% of the gaze data. During such a manual analysis, one typically creates and labels segments in which the subject was looking at relevant objects or items. It is clear that such a manual analysis is a painstaking error-prone task that is extremely time-consuming. Based on our own experiments, we noticed that the annotation of a recording has a time-ratio of at least 10:1, thus one minute of video material takes up a minimum of 10 minutes of annotation. Depending on the level of detail required, this time-ratio may grow up to 50:1. In recent years, some tools were developed to facilitate the manual annotation, e.g. ELAN[1] and ANVIL[2] annotation software. They store the manual annotations in XML-based files, making them transmittable amongst annotators. Despite the fact that manual analysis is time-consuming, it can be seen as the most accurate method since often multiple human annotators are involved in the analysis of the same recording to cross validate each other's annotations. Furthermore, manual analysis is applicable to any object or item of interest including persons and even relevant body parts.

[1] https://tla.mpi.nl/tools/tla-tools/elan/
[2] http://www.anvil-software.org

2.5.2 Marker-based analysis

Apart from manual analysis, the best-known technique is the use of markers to predefine potential Areas Of Interest (AOI). These systems, which either use physical infra-red markers (e.g. Tobii Glasses) or natural markers (e.g. SMI Eye Tracking Glasses), determine the boundaries of the Areas Of Analysis (AOA), generating a two-dimensional plane as shown in figure 2.11, within which eye gaze data can be collected for longer stretches of time and generalised across subjects. The output of this type of analysis is often represented in heat maps or opacity maps that highlight the zones within the AOA that received most visual attention (measured in terms of visual fixations and fixation times). Despite their advantages in comparison to manual analysis, marker-based systems suffer a range of limitations including the need for a fixed position of relevant objects to be tracked. The marker-based approach, for example, is applicable to gain insights into the visual behaviour towards a shelf,f as shown in figure 2.11. However, if a subject grabs a product from the shelf, the benefit of the marker-based approach is lost, since the product is no longer within the 2D-plane. Furthermore, multiple identical objects need multiple markers. When for example an eye-tracking experiment is conducted to gain insights into visual behaviour towards exit signs during a fire drill, markers should be fitted around each individual exit sign. Another important disadvantage is that the AOA should be defined before the experiment, which implies that the automatic analysis is restricted to these regions. Nevertheless, it might happen that afterwards one is interested in the visual behaviour to other regions as well. To overcome this problem one must either repeat the experiment using the extra AOA's or fall back to manual analysis for these particular regions. These shortcomings impose limitations on the efficient use of mobile eye-tracking in real-life settings with moving subjects, objects and a dynamic environment. More limitations of the marker-based systems are discussed in [23] and [49].

2.5.3 Semantic analysis

An alternative for the marker-based analysis is the so-called semantic gaze mapping. The goal of semantic gaze mapping is to reduce the analysis time of mobile eye-tracking recordings, without the need for additional markers. In such an analysis, the visual behaviour of a participant is analysed automatically using reference images that represent the AOIs. For example, to analyse the visual behaviour in the context of a marketing experiment, in which one is interested in the visual behaviour towards a shelf, one has to provide a reference image of the specific shelf to the analysis software. Then, based on computer vision algorithms, this analysis software is capable of automatically mapping

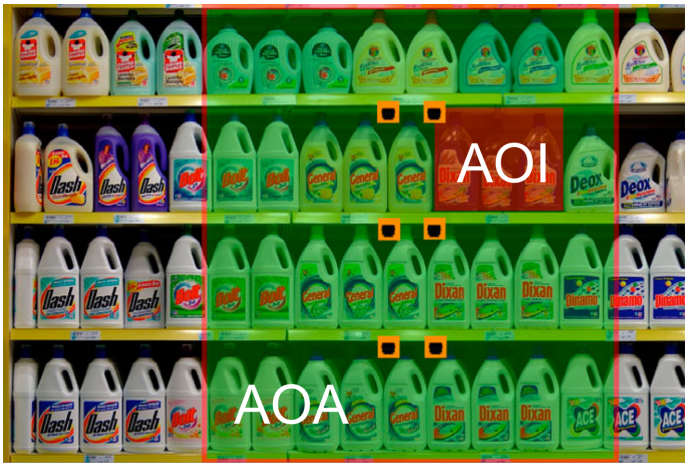


Figure 2.11: Illustration of both AOA (green region) and AOI (red region) in a marker(orange squares)-based analysis approach. Image from [3].

fixations to the image of the shelf. Such an approach is indeed a useful tool that reduces the analysis time significantly. Another advantage of the semantic analysis is that data for multiple participants can easily be aggregated. Common applications in which the semantic analysis has already proven to be a valuable tool include the analysis of visual behaviours towards packages, cockpits, mobile devices, posters, etc. Despite the great potential, the approach suffers some limitations. The most important one is that it is only applicable to analyse the visual behaviour towards objects whose shape does not change during the recordings. Indeed, when one is interested in the visual behaviour towards other persons or faces, the proposed method is inapplicable since the pose of a human may change throughout a recording. On top of that, the purchase price of €10.000 in case of the *Tobii Pro Glasses Analyzer* makes the analysis software expensive. The semantic gaze mapping is also a relatively recent approach. In case of Tobii, they include the semantic gaze mapping since 2015 (three years after we started introducing computer vision techniques in the mobile eye-tracking domain) in their *Tobii Pro Glasses Analyzer* analysis software.

It is clear several attempts were made to facilitate the analysis of mobile eye-tracking recordings. Solutions are available for the analysis of visual behaviour against specific objects. However, they suffer important limitations. For the analysis of visual behaviour towards other persons or towards relevant body parts, such as the face on the other hand, no automatic approaches exist, thus here manual analysis is often the only option. This definitely proves that there is still room for improvement in the analysis of mobile eye-tracking experiments.

2.6 Application domains

In this section, we give an overview of applications in which mobile eye-tracking is used. Many of these applications were already examined using commercially available screen-based eye-trackers during the last decades. However, the development of customer available mobile eye-trackers paved the way for a new area of real-life experiments in which new insights can be retrieved. As mentioned above, the full potential of mobile eye-tracking is often restricted by the complex analytical procedure needed to make sense of the data. Therefore, we argue that a series of research fields and applications could benefit from a semi-automatic analysis framework.

Healthcare

Eye-tracking is one of the methods that provide a better understanding of cognitive processes such as decision-making and problem solving. In [55], mobile eye-tracking is used to measure how infants employed gaze while navigating obstacles, manipulating objects, and interacting with mothers. Results revealed new insights into visually guided locomotor and manual action and social interaction. Another application of mobile eye-tracking involves studies on autism, since it is well known that atypical patterns of gaze and eye contact have been identified as potential early signs of autism [139].

Driver safety

In research on driver safety, there is a large interest in what catches the eye during driving: road signs, vulnerable road users, road crossings, etc. Research on driver safety has been done for several years, however mainly in driving simulators using screen-based eye-trackers. Mobile eye-trackers allow these experiments to be performed in an actual car in real traffic [74]. In these real-life experiments tasks such as obstacle avoidance and navigation are indispensable as compared to simulated environments.

Sports and kinematics

In sports psychology, there is growing interest in mobile eye-tracking since it provides insight into attentional focus, trajectory estimations, visual search strategies, and eye hand coordination. The visual behaviour of experienced athletes reveals specific search strategies and trajectory-estimation skills, which

can play a vital role in talent recognition. By interpreting such recordings, trainers can give better feedback on what is done incorrectly. Next to individual visual behaviour, eye-tracking can also reveal how team members interact during activities [90].

Market research

A well-known application of mobile eye-tracking is found in shopper research. Here, mobile eye-tracking offers an objective measurement of how well products catch the eye in a shop. Market researchers and (brand) developers benefit from insights into the effectivity of point-of-sale displays or the effect of package design, shelf placement, and store planning on shopper experience. The specific context of a supermarket presents a series of challenges, for example a multitude of objects with different shapes and colours, products presented in groups on the shelves or products within the same range exhibiting similar features. As explained above, the limitations of using predefined AOA's makes it virtually impossible to process detailed large-scale shopping experiments beyond a lab-scale one-shelf shop imitation [118].

Customer journey

A prominent field of application for mobile eye-tracking is customer journey analysis. The main purpose of customer journey research is to gain insights into the experience of customers. An example application is buying a train ticket in a railway station. Here, researchers are interested in the entire route and experience of the customer, starting from entering the railway station, to finding the way to the ticket counter, interacting with the teller, finding the correct platform, asking directions to an officer and finally entering the train. Mobile eye-trackers provide potentially useful information on customer experience, particularly when the paradigm is combined with other sensors, such as wearable Electroencephalography (EEG) devices [11]. Customer experience can be measured by using so-called touch points, the contact moments between the customer and the company e.g. in the case of advertising, communication with desk members, etc. The recordings of the mobile eye-tracker can be used to analyse the visual behaviour towards the physical and human touch points such as human contacts and visual behaviour towards specific objects.

Human-human communication

Another application is the use of mobile eye-tracker data for human-human communication experiment. Recent research on multimodal human-human communication has explored the role of gaze in turn taking and feedback in face-to-face conversation [68], shared gaze in dialogue and the function of gaze as a directive instrument in communication [22]. Next to its use in basic research, the study of the distribution of visual attention during communicative interaction is of particular relevance to professionals in training and consultancy (e.g. presentation and meeting skills, sales training, etc.). Among the questions that are addressed in this field are: Does a speaker visually address his/her audience during a presentation? How does the audience divide its visual attention between a speaker and relevant artefacts such as the projection screen? Do gestures of the presenter influence the visual behaviour of spectators?

Musical interaction

Screen-based eye-tracking technology has found its way into music performance research in studies on music reading [43] for several years. More recently in 2015, mobile eye-tracking was introduced within this research field by Vandemoortele et al. [132]. Here the focus lies on the function and timing of interactive gaze behaviour between ensemble players. Using mobile eye-trackers allows researchers to focus on solitary (one musician looking at the other one) and mutual (both musicians looking at each other at the same time) gaze events in musical duos, which was unknown territory until then.

Wayfinding

Wayfinding includes various methods in which people orient themselves in a physical space and navigate from place to place. Research on wayfinding has been done for several years using screen-based eye-tracking and navigating in a virtual reality (VR) building. In 2013, Schwarzkopf [117] discovered differences between VR wayfinding and real-life wayfinding. The cause of these differences is found in the fact that a larger amount of sensory input is perceived in real-life and that, when actually walking through a building, the visual perception of a scene changes continuously. Due to these differences, mobile eye-tracking is considered to be the best way to measure the efficiency of signs in public buildings [27] (e.g. airports, train stations, hospitals, etc). Besides indoor applications, mobile eye-tracking can be used outdoors as well. For example, the use of mobile eye-trackers to get insight in the visual attention of bicyclists on

both good and bad biking trails [133] or evaluating the efficiency and usefulness of fire exit signs during a fire drill.

Usability research

A classic example of usability testing is found in website testing using screen-based eye-trackers. Advancements in mobile eye-tracking open the door to new types of studies that have not been possible previously due to cumbersome technology. Now researchers can equip subjects with eye-tracking glasses and better understand how a person interacts with different messages and channels in any environment. Mobile eye-trackers allow to discover how users interact with apps on smartphones and tablets and how people use their mobile devices as second screen besides the traditional TV.

It is clear that mobile eye-tracking is used in a variety of applications, each with its own challenges and relevant objects of interest. The goal of this PhD thesis is to develop a framework that (semi-)automatically provides calculations for how often and how long the subject is looking at objects, other persons, gestures, etc. It is important that such a framework does not impose restrictions on the real-life aspect of mobile eye-tracking, as is the case with marker-based analysis in which the flexibility is restricted.

2.7 Recorded datasets

Given the broad range of mobile eye-tracking applications, we conducted several experiments in order to test and validate our algorithms on relevant recordings. Throughout this PhD research, we were involved in other eye-tracking studies as well. In this section we describe the most important recordings that we made. Various recordings were used for validation purposes.

Wayfinding

Wayfinding in real buildings is a common application of mobile eye-tracking. To test our semi-automatic analysis on this type of recordings, we conducted a small scale wayfinding experiment in a university building in Antwerp (KU Leuven, Campus Sint-Andries). During these experiments, different scenarios were addressed. First, the subject was instructed to find the way to the library while paying attention to the available signs. After arriving at the library, the subject was asked to search for three specific magazines and books. Besides

this indoor experiment, we also performed an outside experiment in which the subject was instructed to find the way to the local super market starting from the university building while paying attention to traffic sings. Unfortunately, the gaze data of the latter experiment was not usable due to interference between bright sunlight and the IR illuminators. Both experiments were recorded using the older Arrington Gig-E60 eye-tracker, which was connected to a battery pack and laptop in the backpack of the subject. The total duration of these experiments was 14min. 56sec. An example frame of this type of experiment is found in figure 2.12(a). In the analysis of such an experiment, one is typically interested in how frequent the subject looked at specific signs during the entire experiment.

Lecture recording

As mentioned above, there was a close cooperation between our research group (EAVISE) and the MIDI research group during this entire PhD research. One of the main research topics of MIDI involves the interaction between a speaker and its audience. Questions to be answered within this research field are: How does the audience divide his visual attention between a speaker and relevant artefacts such as the projection screen? Do the gestures of a speaker influence the visual attention of the audience? To provide answers to these questions, different eye-tracking recordings were made of students attending a lecture. In total seven students were recorded during four difference lectures. Both Arrington Gig-E60 as well as our Pupil-Pro eye-trackers were used in these experiments. In total several hours of video material were recorded. An example frame of one of these recordings is found in figure 2.12(b).

Customer journey analysis

Another application that we addressed in our recordings is the customer journey analysis. We performed a large scale customer journey experiment in which a user experience bureau was involved (Monkey Shot[3]). The recordings were made in Museum M in Leuven and focused on the experience of visitors of a specific *Hieronymus Cock* exhibition. In total, 14 subjects participated in this experiment. Each participant was equipped with a mobile eye-tracker before entering the museum. They were instructed to buy a ticket at the front desk and then they had to ask directions to the exhibition. After spending approximately 30 minutes at the exhibition the recordings were ended. Questions to be answered from these recordings: Which way visitors take to get to the

[3] `http://www.monkeyshot.be`

exhibition? Do they use the elevator? How often do they look at the walking guide? Do they notice specific works of art, etc. Five eye-tracker devices were used for this experiment: two Arrington Gig-E60 eye-trackers, two Pupil-Pro devices and one Tobii Glasses 1. In total over 500.000 images or 6 hours were captured during this experiment.

Human-human interaction

Another main research field of the MIDI group involves research on human-human interaction in natural settings. Here they are mainly interested in the role of gaze during conversations. Questions to be answered include: does one visually address an interlocutor during speaking or does a listener pay visual attention to the speaker. In such an experiment, in which often two or three participants are involved, a mobile eye-tracker is used by each participant. Furthermore, an additional external camera perspective is added to get an overview shot in combination with a microphone to record the audio as well. This complex setting causes additional difficulties in the analysis apart from the immense amount of data such a recording generates. It is of vital importance that each video and audio stream is synchronised, since we are interested in specific visual behaviour as response to someone else's actions or spoken sentences. When the recordings were made using fixed frame rate devices such as Arrington Gig-E60 or Tobii glasses, the manual syncing operation is manageable using specific software such as Adobe Premiere Pro. However, using our Pupil-Pro eye-trackers, which are webcam-based, a fixed frame rate is not guaranteed making the manual synchronisation extremely complex.

To overcome this synchronisation issue, we developed a small program to perform this synchronisation automatically. Given at least one video stream with a fixed frame rate (which is always available by the external camera), our program manages to synchronise the remaining videos. This is achieved by mapping the available timestamps of the Pupil-Pro eye-trackers onto the timestamps of the external camera. In case particular frames are missing from the Pupil-Pro recordings, our software automatically fills in these gaps by repeating the previous frames. As part of this automatic synchronisation we automatically combine the video streams into one combined stream. An example of this combination is found in figure 2.12(c). The bottom right part shows a frame from the overview camera. The remaining images are the corresponding eye-tracker images. This automated synchronisation offers a significant reduction in manual workload.

Musician rehearsals

A last project in which we participated during this PhD research is the *Into The Wild* project. Here the focus lies on gaining insights into the visual behaviour of musicians in ensembles. Five duos were recorded while playing and working on a piece of their choice. The instrumentation of each duo was unique, ranging from relatively unchallenging (two flutes, two guitars) over moderately challenging (harp-violin, clarinet-piano) to challenging (two percussionists) regarding to the implementation of mobile eye-tracking. Each duo was recorded during two or three rehearsal sessions. Questions to be answered in this project are: Do solitary and mutual gaze events tend to reoccur at the same places in the musical piece? Do these events correlate with specific musical characteristics? Do they correlate with specific problems in the rehearsal process? Two Pupil-Pro eye trackers recorded the eye movements of both players during the entire rehearsal session. Furthermore two external cameras were used to get an overview shot in opposite directions. In total, more than 20 hours of mobile eye-tracking recordings were made in this project. Again, we used our automatic synchronisation tool for combining each video stream. An example frame of this combination is given in figure 2.12(d). The top row contains both eye-tracker images, the bottom row contains the images from the overview cameras.

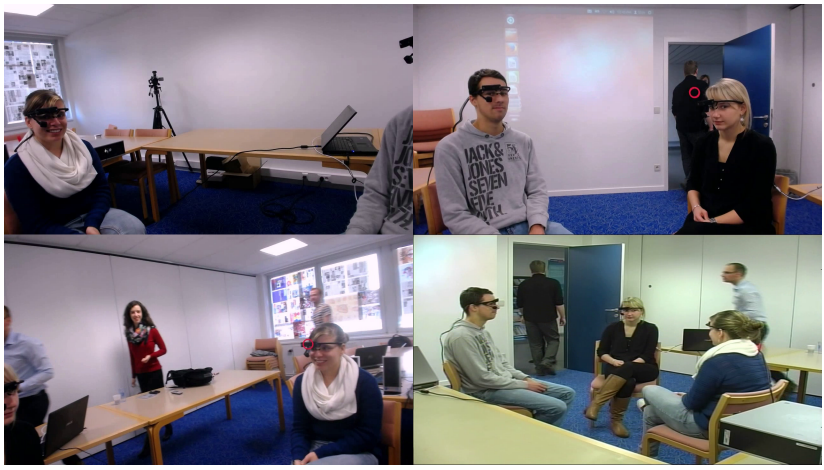
2.8 Challenges

From the application examples described above, it is clear that there is a need for automating the analysis of mobile eye-tracking recordings without restricting the full potential of these real-life situations. In this section we give an overview of the main challenges that we need to tackle in order to develop a (semi-) automatic framework for the efficient analysis of mobile eye-tracking recordings. Several of these challenges will be addressed in the following chapters of this dissertation.

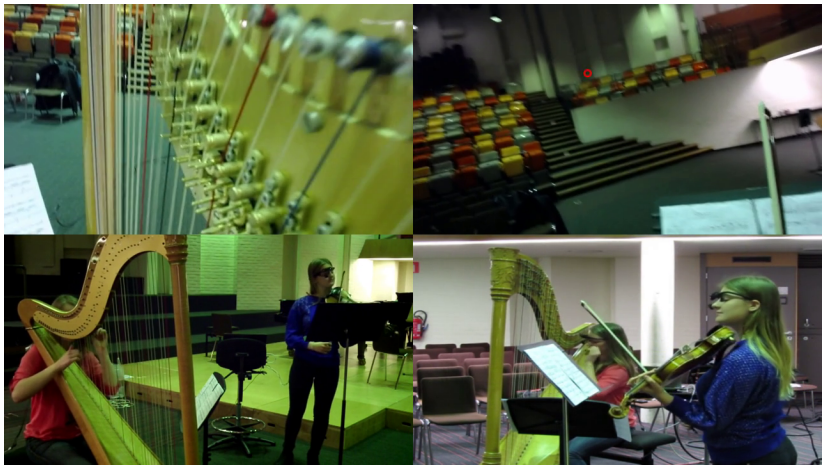
Firstly, since the analysis of eye gaze data is often the first step in more advanced analysis, it is important that this initial step is highly accurate. In current manual analysis protocols, a recording is often analysed by multiple human annotators to remove erroneous annotations. In order to compete with this manual analysis, our approach should be as accurate as possible. To achieve this goal, we need to make well-informed choices regarding the employed computer vision algorithms. However, real-time performance is not an issue in this application since we tackle the post-processing of the recordings. Our technique will only be valuable if it takes significantly less time and labour than manual analysis.

(a) Example of wayfinding recording

(b) Example of lecture recording



(c) Example of our automated synchronisation for human-human interaction experiments



(d) Example of our musicians recordings in which two eye-trackers and two external cameras are used

Figure 2.12: Example frames of various mobile eye-tracking recordings that were made throughout this PhD.

Secondly, the images that we process are recorded by a head-mounted device, i.e. the scene camera of a mobile eye-tracker. This implies that we encounter six degrees of freedom in the position of the camera. On top of that, we will face rapid movements due to the natural behaviour of humans. Since the background is highly dynamic, we are unable to employ often used background segmentation techniques as a basis for our processing steps. Instead, we need to rely on more complex approaches to achieve our goal. Apart from a moving camera position we also have to deal with moving items or subjects in the scene.

Third, our goal is to develop a generic analysis framework applicable to each type of mobile eye-tracker. This implies that our algorithm should be applied on recordings of various resolutions and frame rates.

Finally, we opted to focus on four main topics within our semi-automatic analysis, each with its own specific challenges.

1. Object recognition: depending on the used eye-tracker, the resolution of relevant objects in the images is sometimes very low, making the object recognition challenging.

2. Person detection: due to the natural setting we encounter, persons sometimes appear highly deformed in the images captured by the scene camera, in particular when looking at a person gesticulating. On top of that, due to the specific camera angle and position, people are often not visible from head to toe, making some existing detection approaches inapplicable.

3. Hand detection: depending on the experimental setup, the distance between the observer and the participant is often large, making the hands only a small fraction of the entire image. On top of that, the natural character of our experiments often causes rapid hand movements, which introduces motion blur. The combination of both low resolution and fast movements makes it extremely hard to detect and track hands in long-lasting recordings.

4. Gesture detection: again, fast moving hands sometimes make it difficult to keep track of them. Furthermore, gestures often consist of complex motion trajectories. Since we offer the speaker the ability to move freely, it becomes hard to disambiguate hand movements from body movements.

2.9 Conclusion

This chapter motivates why this doctoral research is needed. We gave an overview of existing eye-tracking approaches as well as a detailed overview of video-based eye-tracking including screen-based and mobile eye-tracking. Although it is clear that mobile eye-tracking offers more flexibility, the analysis of the data is far more complex. We extensively discussed existing approaches for the analysis of mobile eye-tracking recordings, indicating their limitations for real-life mobile eye-tracking experiments. Based on the large variety of mobile eye-tracking applications, we defined a set of main topics which will be addressed throughout this dissertation.

First we will focus on the detection of specific objects in images captured by the scene camera of a mobile eye-tracker to automatically count how often and how long the subject looked at them. Such an approach is of great importance in for example wayfinding experiments, in which visual behaviour towards various signs is of key importance, or for marketing experiments, in which one is interested in visual behaviour towards specific brands. In a next phase, we focus on the detection of humans in these images in order to automate the analysis. This is useful in the analysis of customer journey experiments, which are often long-lasting recordings, and therefore practically infeasible for manual analysis. In a third step, we focus on the detection of human hands in images to automate the analysis of visual behaviour towards finer body parts. This is in particular useful for the analysis of recordings that are made within the field of human-human interaction research. Detection of hands in images is a first step in gesture analysis, which is further explored in the fourth part of our approach. Analysing the visual behaviour towards gestures is important in for example studies on presentational or communicative skills.

Thus, our goal is to develop a framework for the semi-automatic analysis of mobile eye-tracking data, allowing a more efficient analysis which is less time-consuming and requires only a fraction of manual workload. However, developing such a framework is a far from trivial task due to the challenges related to the analysis of mobile eye-tracking recordings. Which are: moving camera position as well as moving objects within the scene. Furthermore, these movements often occur fast, which causes motion blur, making the image analysis complex.

In the next chapters, we thoroughly discuss each topic of our analysis framework starting with the object recognition in chapter 3.

Chapter 3

Object recognition

In this chapter we propose the first part of our analysis framework, i.e. an automatic object recognition approach. Such an approach allows us to automatically detect specific objects in images captured by the scene camera of a mobile eye-tracker. By mapping the gaze data on top of the detected items, we get insights into the visual behaviour.

This chapter is subdivided into five main parts. Section 3.1 gives an introduction on our object recognition approach in the context of mobile eye-tracking. In section 3.2, an overview of existing object recognition approaches is given. Section 3.3 describes our approach, while in section 3.4 we explain the integration of manual interventions to further increase the accuracy of our system. Finally, in section 3.5 the results of our object recognition approach are discussed.

The work presented in this chapter was published at the SAGA 2013 conference [35] and at the VISAPP 2014 conference [36].

3.1 Introduction

Eye-tracking experiments are often performed to measure how often and for how long a subject looked at a specific object (or part of an object), to gather information about what 'catches the eye' in a certain setting. Well-known applications in which visual behaviour towards specific objects is of vital importance include marketing, where one is interested in the visual attention towards products of a particular brand and wayfinding experiments, where the efficiency of signage is validated. As explained in chapter 2, a common approach

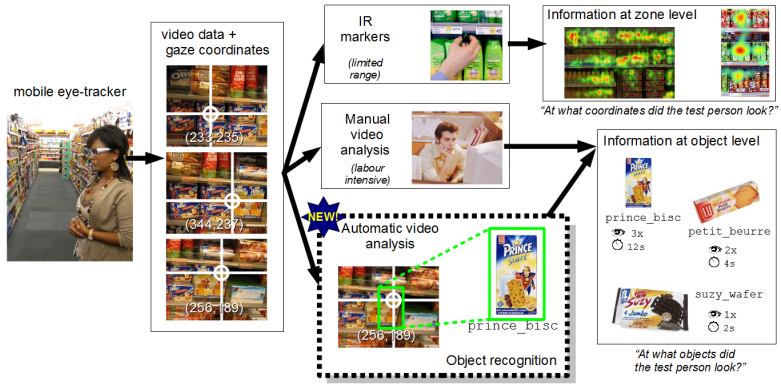


Figure 3.1: High-level overview of current analysis methods and how our approach fits between them.

for this type of analysis is the use of (IR)-markers to predefine potential AOI. In this chapter we present an alternative to these AOI-based methods, building on recent studies combining several image processing techniques with eye-tracking data [125, 142]. By mapping gaze data on objects to be recognised in the scene video data, a number of restrictions of AOI-based approaches no longer hold, including the need to work with predefined static areas or virtual 2D reference frames. Objects for which gaze data statistics need to be generated can be selected in the actual video stream, without prior training.

The schematic representation in figure 3.1 gives a clear overview of both existing approaches and our computer vision-based approach. The analysis of a mobile eye-tracking experiment is currently done using two main methods. The marker-based approach offers an automatic analysis and answers the question 'at which **coordinates** did the subject look?'. The manual approach on the other hand, which is labour intensive, answers the question 'at which **object** did the subject look?'. Our object recognition based analysis combines the advantages of both approaches since it automatically provides information at the object level. Starting from images from the scene camera and a list of associated gaze data, we use an image recognition algorithm to count how often and for how long the subject looked at a specific object or item.

3.2 Related work

Object recognition, or finding an object that is identical to a trained one, is traditionally realised with *local feature matching techniques*. Recognition methods define local interest regions in an image, based on specific features of the image content, which are described with descriptor vectors. The characterisation of these local regions with descriptor vectors that are invariant to changes in illumination, scale and viewpoint enables the regions to be compared across images. Differences between approaches lie in the way in which interest points, local image regions, and descriptor vectors are extracted.

An illustration of a basic object recognition task in the context of an eye-tracking experiment can be found in figure 3.2. Suppose one wants to find out whether the sign is present in the right image. The left image is a reference photo of the sign to be recognised. In a first step (as can be seen in the middle row) features are extracted in both images, illustrated by the coloured circles. In a second step, as can be seen in the lower part of the figure, the object recognition algorithm searches for similar features in both images as illustrated by the blue lines. Based on the number of correspondences, their confidence and relative positions, one can decide whether the sign is present in the second image or not.

Many object recognition algorithms based on this technique have been proposed. A survey is given in [127], while [95, 96] report on comparative experiments. An early example is the work of Schmid and Mohr [115], where geometric invariance was still under image rotations only. Scaling was handled by using circular regions of several sizes. Lowe et al. [87] extended these ideas to real scale-invariance in his widely adopted Scale Invariant Feature Transform (SIFT). More general affine invariance has been achieved in the work of Baumberg [13], Tuytelaars & Van Gool [128, 129], Matas et al. [91], and Mikolajczyk & Schmid [94].

In recent years, the development focus of this field shifted from accuracy to computational efficiency. In order to reduce the computation time of SIFT, many improved versions were proposed, such as PCA-SIFT [78], FAST [112] and SURF [14]. By using integral images and box filters, SURF reduces the computation time and improves the speed of detection. Moreover, SURF's detector and descriptor are not only faster, but the detector is also reported to be more repeatable and the descriptor more distinctive than SIFT.

Although SURF and SIFT showed their potential in a wide range of computer vision applications, a possible shortcoming is that these techniques are not fully robust to affine deformations. When 2D or 3D objects are compared, recognition results are poor if the rotation is extreme or when the viewing angle changes radically. Inspired by the affine invariant techniques of Tuytelaars et

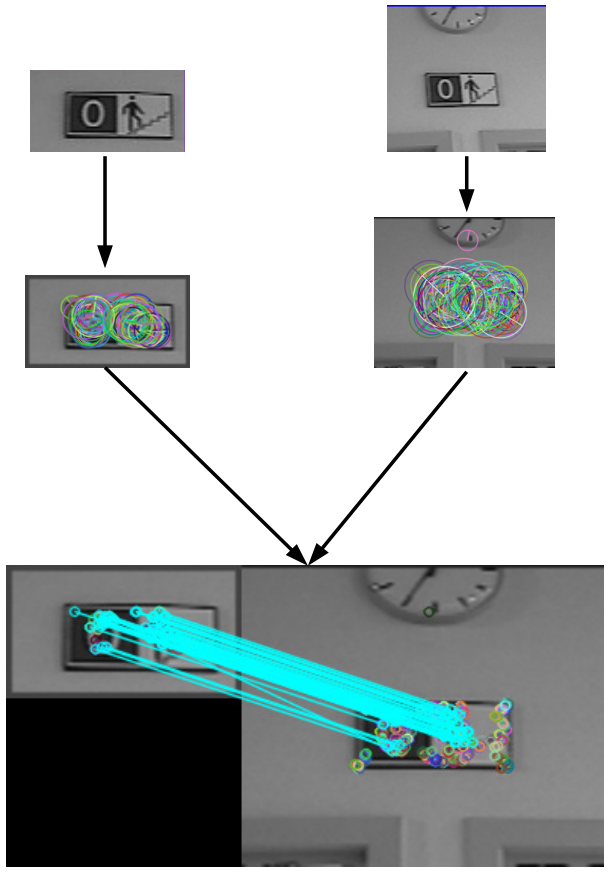


Figure 3.2: Illustration of basic feature matching. Coloured circles represent the features, blue lines represent the matching feature pairs across both images.

al. [128] the full-affine versions ASIFT [100] and FAIR-SURF [103] were recently developed.

Although SIFT and SURF are regarded as state-of-the-art, we were forced to opt for some more recently developed techniques due to licensing regulations. Therefore we compared two competitive alternatives for SIFT and SURF, namely Oriented FAST and Rotated BRIEF (ORB) [113] and Binary Robust Invariant Scalable Keypoints (BRISK) [86].

The ORB feature descriptor is built on the well-known FAST keypoint detector [112] and the recently developed BRIEF descriptor [25]. ORB is

Table 3.1: Experimental comparison of local feature extraction methods.

	ORB	SIFT	SURF	BRISK
Avg pct. inliers	**18.53%**	3.53%	4.46%	5.4%
Avg number of features	496	**3244**	1589	689
Image retrieval success	66.67%	83.3%	**100%**	66.67%
Avg feature detection time	151ms	3417ms	480ms	**52ms**
Avg total time	245ms	4571ms	693ms	**159ms**

a computationally efficient replacement for SIFT and SURF, since it has similar matching performance and is even less affected by image noise. ORB is suitable for real-time performance since it is faster than both SURF and SIFT. Another competitive approach to keypoint detection and description is Binary Robust Invariant Scalable Keypoints (BRISK) as it is as performant as the state-of-the-art algorithms, but with a significantly lower computational cost.

We performed an illustrative experiment in which we compared various local region matching techniques on a set of representative images as shown in figure 3.3. The purpose of this experiment was to retrieve an object in images that are rotated and affine transformed. An overview of these experimental results is displayed in table 3.1. *Avg pct. inliers* stands for the ratio of inliers versus outliers as obtained by a Random sample consensus (RANSAC) [54] validation. *Image retrieval success* represents in how many images the respective object was found. *Avg feature detection time* stands for the average time that was required to calculate the features, while *Avg total time* represents the total time including the feature matching. *Avg number of features* stands for the average amount of features that were found in the images. Without doubt, BRISK is the fastest method, whereas SURF is the most accurate one. ORB on the other hand achieves the highest ratio of inliers versus the total amount of matches, whereas SIFT finds the largest amount of features.

Another evaluation of these detectors is presented in [97]. Although their results demonstrate that BRISK outperforms ORB, we prefer to use ORB in our application based on our own experiments. Mobile eye-trackers are sometimes equipped with low-resolution scene cameras, for example 320×240 pixels on the Arrington mobile eye-tracker. In addition, we are only interested in a specific region around the gaze cursor, yielding a final Region Of Interest (ROI) of maximum 250×250 pixels on images of recent, high resolution, mobile eye-trackers. We noted that applying BRISK to such small images often results in an insufficient number of extracted keypoints, and thus does not generate an adequate number of matches, limiting the applicability of our system. The results of a small experiment are shown in figure 3.4. Here, we show the average

Figure 3.3: The leftmost image includes the object that needs to be retrieved in the other images.

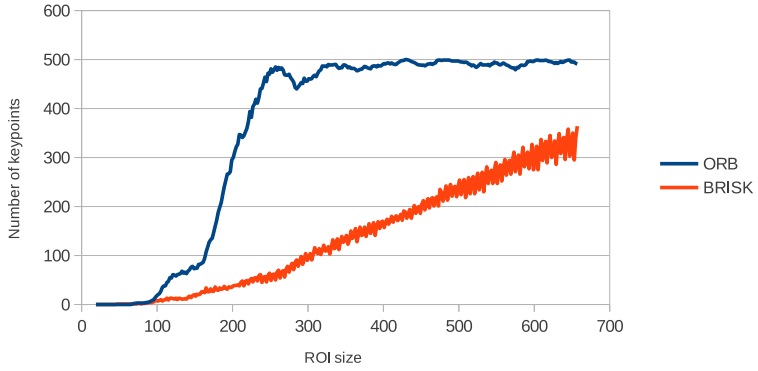


Figure 3.4: Comparison between ORB and BRISK. The horizontal axis represents the size of the ROI square. The vertical axis represents the amount of detected keypoints.

number of keypoints that are obtained by ORB and BRISK on various image sizes. Indeed, this figure reveals that using BRISK on small images often results in an insufficient number of keypoints, whereas ORB retrieves much more keypoints.

Besides an overview of feature detectors and descriptors, we also describe existing approaches that combine object recognition and eye-tracking recordings such as in [19, 64]. However, the evaluation of this integration is not discussed deeply in these works. In [125] on the other hand, more quantitative measurements are given. Toyama et al. developed a novel Augmented Reality (AR) application named Museum Guide 2.0 that utilises eye-tracking as an interactive interface and that recognises objects in a real environment. The basic idea of Museum Guide 2.0 is that visitors of a museum would wear a head-mounted eye-tracker while strolling through an exhibition. Whenever the user looks at any of the exhibits for a certain duration, the system automatically presents corresponding AR meta-information. There is a strong similarity between their goal and ours, i.e. using object recognition algorithms to automatically determine which object a subject is looking at. On the other hand, there are some important differences: they rely on a pre-trained database containing multiple images per object of interest, whereas we would avoid this training step as much as possible. Furthermore, they validate their approach using some toy-examples, which are visually easy to distinguish, placed on a clean table. This makes their experiments not representative, whereas we tackle the analysis of challenging real-life recordings.

Another example in which object recognition and eye-tracking is combined is found in [142]. Here, the potential for combining human and computational input into integrated collaborative systems for image understanding is explored. Rather than applying object detectors at every location in an image arbitrarily, they could be more intelligently applied only at important locations as indicated by gaze fixations. This would not only minimise the potential for false positives, but also constrain the true positives to only the most user-relevant content. Although their approach improves the detection accuracy on the tested images, it's not relevant in our application since their focus is highly different from ours. Their purpose is to improve the classification results on standard image datasets by using gaze information to highlight regions in the images. Therefore, they used a screen-based eye-tracker to record the visual behaviour of participants to various images. Based on that visual behaviour they select the regions in the images that attracted visual attention since they may contain interesting or relevant objects. In a next phase, they apply an object recognition approach only on the selected regions rather than analysing the entire images.

Based on this overview of relevant literature, we opted for an object recognition approach using the ORB feature descriptor. In contrast to existing approaches, we tackle an object recognition approach in which the tedious creation of a database of each object of interest is avoided as much as possible. In the next section we describe the integration of the ORB feature descriptor in our framework.

3.3 Approach

The input of our algorithm consists of a video stream, captured by the scene camera of an eye-tracker, and a data file which contains the corresponding gaze data. It is important to note that our algorithm is able to process both raw gaze locations as well as the actual fixations. Since we only rely on the video stream from the scene camera and a text-file containing gaze data, our approach is independent from the eye-tracking brand that is used, and therefore it maximises the applicability.

As explained above, the task of the object recognition approach is to count how often and how long a subject looked at a specific object. This is accomplished in five steps as explained below:

1. Preprocessing step: since we are only interested in the objects that appear close to the visual fixation point, the input images of the forward looking camera are cropped around the coordinates of the corresponding gaze data. This reduction in pixel data to be processed will reduce the computational cost as compared to searching for a specific object in an entire image. However, a ROI that is too small may reduce the accuracy, since fewer features can be found in these relatively small images. Based on experiments using our Pupil-Pro mobile eye-tracker, which captures images of 1280×720 pixels, we empirically determined to crop a ROI of 250×250 pixels around the gaze cursor. A ROI of 250 pixels wide corresponds to a viewing angle of 17.5° in case of the Pupil-Pro mobile eye-trackers. When using our Arrington mobile eye-trackers, a ROI of 120×120 pixels is cut out, which, in its turn, corresponds to a horizontal viewing angle of 21.4° .

2. In the next step, the user selects objects of interest. We developed a user-friendly Graphical User Interface (GUI) in which the user can replay the eye-tracking recording. While replaying, the user can use the mouse to select objects of interest. One can pause the video and draw a rectangle around the object of interest, after which the video automatically continues. The objects are then stored in an object database, avoiding the tedious task of manually creating such a database with pre-captured training images of the objects, as proposed in for example the Museum Guide 2.0 [125].

3. The third step consists of searching for correspondences between each cropped frame and each frame stored in the database, using ORB features. We apply a matching algorithm, based on the Euclidean distance between

features to find similar keypoints between each image pair. Furthermore, we also apply several filter techniques to eliminate weak or false matches.

First, the distance between the two best matches is evaluated: if this distance is large enough, it is safe to accept the first best match, since it is unambiguously the best choice. Second, a symmetrical matching scheme is used, which imposes that for a pair of matches, both points must be the best matching feature of each other. The last step involves a fundamental matrix estimation method based on RANSAC to remove the outliers. This approach ensures that when we match feature points between two images, we only keep those matches that fall onto the corresponding epipolar lines. An illustration of the keypoint matches is given in figure 3.5. The left side of image a and b contains the reference image of the object to be detected, as selected by the user (in this case corresponding to a presentation screen). On the right side of each image we see a cropped region around the gaze cursor of two frames of one of our lecture recordings. On the right side of figure 3.5(a), a part of the object is visible in the cropped region, and there are seven corresponding keypoints between the two images, as shown by the blue lines. In figure 3.5(b) on the other hand, the object is not visible on the right side, thus no corresponding features were found with the exception of one false match.

4. In the fourth step we assign a score S to each pair of images:

$$S = \frac{\sum_{i=1}^{m} d(k_i, k_i')}{m(\sum_{i=1}^{m} A(k_i) + \sum_{i=1}^{m} A(k_i'))}, \tag{3.1}$$

where k_i and k_i' stand for the ith keypoint of the corresponding images, m is the total number of matches, $A(k_i)$ is the pixel area of the corresponding features, and d is the Euclidean distance between a pair of keypoints. This score S is then used to decide whether a cropped frame exhibits sufficient agreement to one of the frames in the database. A cropped frame is counted valid if the value of S is lower than a tunable threshold value.

5. In the fifth and final step we cluster consecutive similar frames into a 'visual fixation'. We define a visual fixation as a series of images in which the same object was viewed with a minimal duration time. This duration is configurable using a slider since the length of a visual fixation depends on the task the test person is occupied with. This minimal length factor allows us to remove many false detections, since one can assume that a valid visual fixation should last at least 150 ms for example (i.e. 5

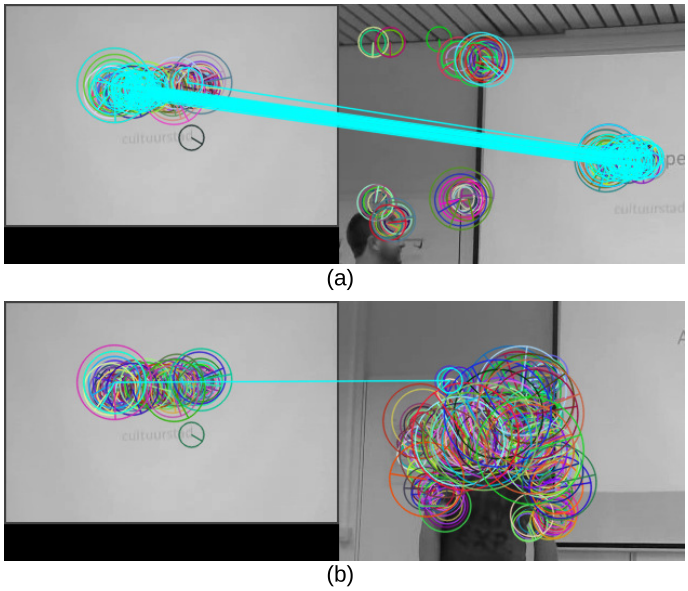


Figure 3.5: Illustration of our feature matching, Blue lines illustrate corresponding features. Part(a) represents a valid feature matching, part(b) represents feature matching in which the object of interest is invisible.

consecutive frames for a 30 fps camera). In case a match between a reference image and a frame from the recording was found in just a single time frame, one can assume that this is an invalid (or too short) visual fixation and therefore it can be discarded.

For each object of interest, our algorithm automatically generates a list of frames in which visual fixations were measured on that particular object. These files can be used for further analysis and visualisation of the eye-tracking recording. In chapter 7, we present the various visualisation methods that we developed. These include, for example, statistical representations in which the user can easily determine which object of interest attracted the most visual attention and the percentage of viewing time of that object as compared towards others. Other visualisations include a timeline or object cloud. Finally, we export the output of our algorithm into a XML-based file, making it integratable with existing annotation files, and thus usable in annotation environments such as ELAN.

3.4 Semi-automatic analysis

Our object recognition approach yields good accuracy, as will be discussed in the next section. However, it might occur that a particular object is hard to recognise. Typically, this occurs when the chosen image of an object of interest is one from an awkward viewpoint as compared to the remainder of the video, or when another object, which is visually similar to the object of interest, is present in the video. To overcome these issues, we propose the integration of manual interventions to further improve the accuracy.

As mentioned above, our approach requires a single image of each object of interest to initiate the automatic analysis of an eye-tracking recording. After this initial processing, it might be useful to manually examine the retrieved ROIs for a particular object, as illustrated in the left part of figure 3.6(a). Here we show our timeline representation in which we zoom in on a specific time slot of the experiment. As indicated by the coloured check marks in this figure, three out of four retrieved ROIs are indeed correct, however the last one is not an image of the given object of interest. By adding the correct samples to the database containing the images of objects of interest, we add multiple viewpoints of the same object. This will further enlarge the chances of retrieving each ROI in which that particular object is visible. On top of that, we can also add the wrongly classified sample by labelling it as a negative sample for that object of interest. The expanded database is illustrated in figure 3.6(b). Here, the coloured squares illustrate positive samples of each object, while the black squares represent negative samples. The coloured circles represent the cropped ROIs that are processed using our algorithm. Based on the Euclidean distance, they are assigned to a specific object of interest. In case the best match of a ROI corresponds to a negative sample of a specific object of interest, our algorithm automatically discards this match and chooses the next best match that belongs to a positive sample. By reiterating the processing of the entire recording using these extra samples, we are able to further improve the accuracy in two ways. The extra positive samples result in additional correctly retrieved ROIs, while the negative samples will reduce the amount of false ROIs. It is important to notice that these manual interventions are optional, and should only be applied in case the accuracy of a particular object is unsatisfactory. However, we see that the accuracy of the analysis increases substantially with a minimal amount of well-chosen manual input. This semi-automatic paradigm will be used in the remainder of this PhD dissertation.

(a) (b)

Figure 3.6: (a) Cut-out of our timeline visualisation in which we can distinguish correct and false object detections. (b) Illustration of the feature space of our expanded database.

3.5 Results

In order to test our object recognition technique, we applied it to a set of image sequences, which were captured by a mobile eye-tracker during the above-mentioned wayfinding recording (section 2.7). This recording was made using an Arrington Gig-E60 mobile eye-tracker, which embeds a scene camera that captures images of 320×240 pixels. As explained in section 3.3, we crop a region around the coordinates of the gaze cursor in each image of the scene camera. Using these mobile eye-trackers, we determined to crop a region of 120×120 pixels, i.e. a horizontal viewing angle of 21.4°. We gathered ground truth by manually labelling a subset of 2000 cropped ROIs around the gaze cursor. Since the objective of the particular experiment was to gain insights into the visual impact of signs in a public building, we only labelled the images in which a sign was visible. This resulted in 1284 frames without a label and 716 labelled frames of five different signs: two emergency exits, stairs, fire extinguisher and toilet.

In figure 3.7 we present the accuracy of our object recognition technique in a precision-recall curve. Such a curve is a frequently used method for presenting the accuracy of object recognition algorithms. The precision (P) is the fraction of retrieved instances that are relevant, while recall (R) is a measure of how many truly relevant results are returned. The mathematical calculation of both (P) and (R) is:

$$P = \frac{T_P}{T_P + F_P} \qquad R = \frac{T_P}{T_P + F_N}$$

Where T_P stands for true positive, F_P stands for false positive and F_N stands for false negative. In such a precision-recall curve, the optimal point is located in the upper right corner, yielding a high precision and a high recall. We

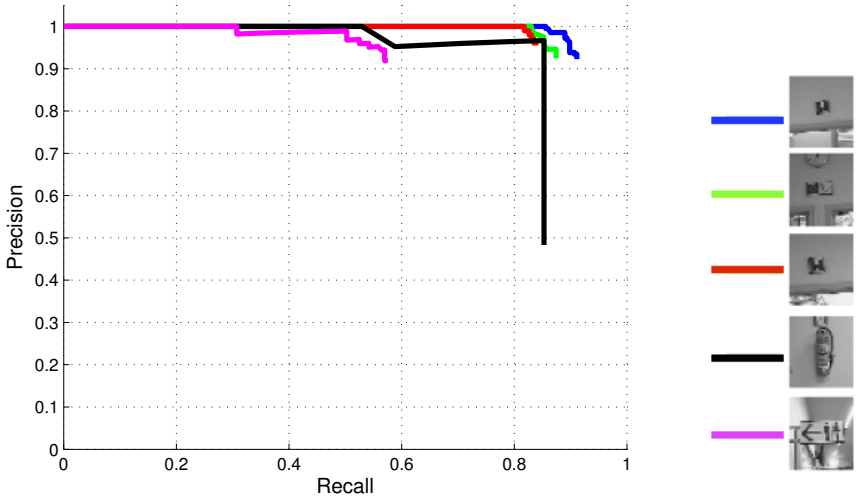


Figure 3.7: Precision-recall curve of our object recognition technique tested on a set of 2000 images. Corresponding objects are shown at the right side of the graph.

created this curve by applying a varying threshold on the score S as calculated in formula 5.5. The obtained detection results are satisfactory for most of the objects. However, the accuracy of the toilet sign is rather low. Because the subject looked at the sign from various distances, whereas the image of this object of interest contains only one viewpoint. As explained above, using our semi-automatic approach, we can easily add another image of this object to further improve the accuracy.

We did an initial experiment on another subset of this recording to prove the usefulness of our semi-automatic approach. Here, we chose an object that was hard to detect throughout a recording. The initial accuracy of retrieving this object is shown by the blue precision-recall curve in figure 3.8. Again, we varied a threshold on score S from formula 5.5 to create this curve. After this initial test, we selected two additional ROIs that were retrieved from the recording and that contain different viewpoints and/or viewing distances as compared to the original image of the object of interest. The blue and red curves show the improvement in accuracy when multiple images of the same object are used.

Since we aim to developed an analysis framework that is faster than manual analysis, the computational cost is, next to the accuracy, also of great importance.

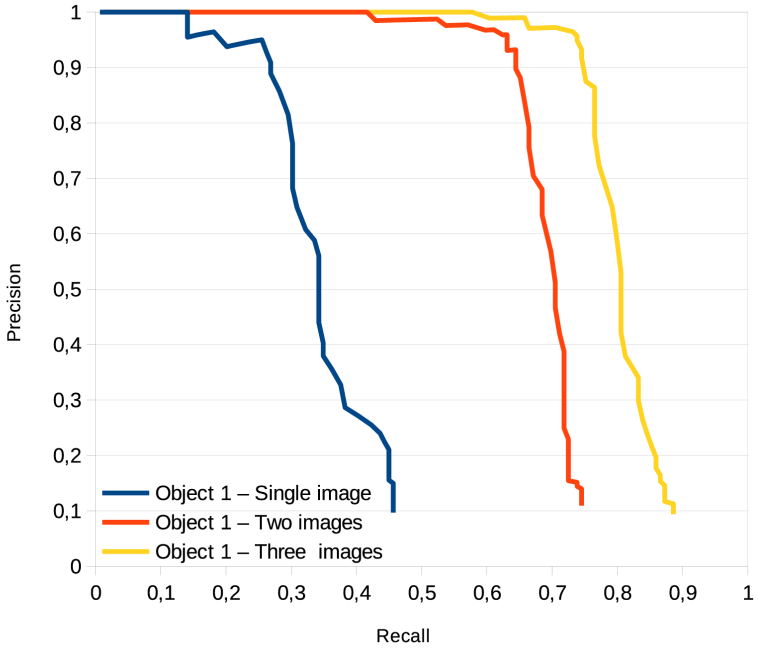


Figure 3.8: Improvement of accuracy when additional correct ROIs are added to the image database.

In table 3.2 we show the processing time for a given number of selected objects and a given number of video frames. As illustrated in this table, data of an eye-tracker experiment of 6000 frames (3m 20s of video data) can be processed in a couple of minutes, less than the duration of the video itself, even when up to five objects of interest are chosen. Remember that manual analysis of such a video takes at least 30 minutes of manual effort. These tests were performed on a normal desktop PC. The ROIs we processed had a resolution of 120×120 pixels. Applying this software to frames with a higher resolution will have an impact on the computational cost since the maximum ROI size equals 250×250 pixels when the Pupil-Pro eye-trackers are used. On the other hand, this approach can easily be implemented on a multi-threaded system in which retrieving the objects of interest in the recording is distributed amongst the available threads.

On top of these prospective experiments, we used the above-mentioned object recognition approach for the analysis of various long-lasting eye-tracking

Table 3.2: Computational time of the object recognition implementation.

# selected objects	2	3	4	5
video of 1m 6s	31 s	42 s	54 s	68 s
video of 2m 13s	61 s	80 s	104 s	133 s
video of 3m 20s	94 s	122 s	162 s	201 s

recordings, which we will thoroughly discuss in chapter 7.

3.6 Conclusion

In this chapter we presented an approach for the automatic analysis of mobile eye-tracker data, based on object recognition. The purpose of this analysis is to automatically provide information regarding the visual behaviour towards specific objects of interest in terms of how often and how long the subject looked at them. As opposed to [125] we presented an object detection scheme in which a separate training step is no longer required in advance. In our approach, it is unnecessary to capture specific images of the objects of interest in advance, since we developed an interface in which images of objects of interest are simply selected from the recording itself. By developing a semi-automatic detection approach, we do allow adding multiple images of a specific object of interest in case the initial accuracy is unsatisfactory.

Based on the analysis of an actual eye-tracking experiment in the context of wayfinding, we prove the usability of our approach. Our method is highly accurate, while the computational time is much shorter than the time required for manual analysis. Since our approach processes each frame of a recording, we are able to generate statistics and other useful visualisations of an experiment. For more information regarding these visualisations we refer to chapter 7.

Chapter 4

Person detection

This chapter describes the second part of our analysis framework: i.e. an automatic person detection approach. Our approach automatically counts how often and for how long the subject looked at another person or more specifically the face of another person. To achieve this goal, we embed several computer vision aspects into our framework, including a human upper body model that we trained in combination with a face detection method. Furthermore, we apply a gaze-based temporal smoothing approach for improving the accuracy.

This chapter is subdivided into 7 main parts. In section 4.1 we discuss applications that can benefit from an automatic person detection system and we explain the shortcomings of existing analysis methods. In section 4.2 an overview of state-of-the-art person detectors is given. Section 4.3 describes each aspect of our approach, while in section 4.4 the integration of manual intervention is introduced. We propose an integration of person re-identification in our framework as described in section 4.5. Finally, in section 4.6 we present accuracy measurements of our approach validated on real-life mobile eye-tracking recordings. A concluding summary of this chapter is given in section 4.7.

The work presented in this chapter was published at the SAGA 2013 conference [35] and at the VISAPP 2014 conference [36].

4.1 Introduction

As presented in the previous chapter, our automatic object recognition approach simplifies the analysis of many real-life mobile eye-tracking experiments, making

it a valuable alternative for the marker-based approach. On the other hand, several studies in the field of visual behavior have shown that visual attention is particularly attracted to other persons [71, 131] and faces [62]. Since each person and face is unique, we cannot use our object recognition approach since the task of object recognition consists of retrieving a given object that is identical to a trained object in a set of images. To overcome this problem, another range of image processing techniques exists, viz. object detection algorithms. Here, the principle of detecting objects with a known specific appearance is extended towards detecting objects based on a general object class model that contains intra-class variability. In our automatic analysis, we use object detection techniques for detecting human bodies and faces in images, as these are two kinds of object classes of which the appearance varies a lot within the class[1].

Similar to the object recognition approach, we analyse the images captured by the scene camera of a mobile eye-tracker. Here, we search for persons and faces within these images using an object detection algorithm. In case a person is found, we check whether the gaze cursor overlaps with this person detection. Using such an approach, we are able to count how often and for how long the subject looked at another person. Furthermore, we integrated a person re-identification algorithm in our approach, enabling the automatic retrieval of information on which person the subject looked at.

It is clear that the existing marker-based methods are inapplicable in this type of experiment since it is cumbersome or even practically infeasible to fit IR-markers on human bodies. Therefore, in practise, the analysis of this type of recording is currently done manually. It is evident that by introducing our automatic person detection approach into this field of research, a significant amount of manual labour can be reduced.

In the context of eye-tracking experiments, such a person detection approach is of particular importance in two main applications. A first application field is customer journey experiments. Here, the experience of customers is measured using so-called touch points i.e. the contact moments between the customer and the company. Often these contact moments involve human contacts e.g. asking information at an information desk, asking directions, etc. Our person detection approach can be used to provide a first step in the analysis of these recordings viz. by automatically answering questions such as *when and how often does the subject look at a person?*. This restricts the manual analysis to merely interpreting these automatically generated fragments, which is the least time-consuming part of the analysis.

[1]It is important to notice that in the context of image processing a person is also seen as an 'object' to be detected in images.

Another prominent field of application in which our person detection is useful, is the analysis of human-human interaction experiments. Here, recordings are made of one or multiple subjects wearing mobile eye-trackers during specific tasks, such as attending a presentation or a natural conversation amongst subjects. The purpose of these recordings is to get insights into visual behaviour towards other persons. It is clear that a crucial part of the analysis consists of determining when and how long the subject looked at another person, being the presenter or an interlocutor. Once this initial information is retrieved, researchers may search for relationships between speech and visual behaviour or visual behaviour related to gestures of the interlocutor. Again, our automatic person detection approach can be used to provide these initial analyses, i.e. identifying when, how often and for how long the subject looked at another person.

4.2 Related work

First, we introduce the concept of object detection since it is a general approach that is applicable for numerous other applications besides person detection. Next we give an overview of existing state-of-the-art person detection algorithms. Finally, we motivate why we opted for specific algorithms in our approach.

General object detection

Typical in object detection, a model is trained using both positive (images that contain the concerning object) and negative (images that do not contain the object) samples. To detect a specific object, a pre-trained model of the object to be detected is searched for over the entire image. Since the object may appear in any size in the image and the model is often fixed-sized, it is advisable to perform this evaluation over a range of scales. This is mainly achieved using a scale-space pyramid in which the input image is downsampled several times with a specific factor. Traditionally, the image is downsampled until its size corresponds to e.g. the height of the model.

Once the scale-space pyramid is constructed, several features are calculated for each layer of this pyramid, resulting in a feature pyramid. Often, these features are commonly calculated on specific positions based on a discrete step size of a few pixels. This step size depends in turn on the scale at which the features are calculated. These features include edge extraction or Histograms of Oriented Gradients (HOG), etc.

The evaluation of the image, i.e. searching whether a specific object is present in an image, is often achieved using a sliding window approach in which the pre-trained model is shifted over various positions in each layer of the feature pyramid. At each evaluated position a comparison is made between the calculated features and the model. The way this comparison is done depends on the specific detector that is used. Often the detection model is a machine learning classifier such as a support vector machine (SVM) that classifies and scores each evaluated position. This score can be seen as a measure of the likelihood that a specific image position contains the object of interest. By applying a threshold to this score, one can set a specific working point for a detector. Using a low threshold, each object will be retrieved, however with the risk of generating lots of false detections (hence high recall at low precision). A high threshold on the other hand will retrieve less objects, but these will mostly be correct ones.

Typically, candidate object locations are indicated using a bounding box. Since the sliding window approach is often used in these algorithms, it is common that multiple overlapping detections are found around the same object location. To overcome this issue, an additional non-maxima-suppression (NMS) is performed. When multiple bounding boxes overlap each other more than a specific overlap criterion (e.g. 50%), only the bounding box with the highest score is kept.

These object models are constructed in an offline phase. For this, based on both positive image patches (those containing the object of interest) and negative patches (random patches with e.g. backgrounds not containing the object of interest) a model is trained as follows. Features are calculated on each patch and are fed into a machine learning classifier. The goal of this classifier is to find an optimal detection model able to generalise to instances of the object not present in the training set, while still being specific enough to only detect these objects of interest. When a proper classifier is trained, it is capable of retrieving an object in images despite changes in viewpoint, illumination or appearance. It is important to mention that one can train a model of any object, as long as there is sufficient variation in views. Examples in which object detection is useful include: fruit detection, car detection, etc.

Person detection algorithms

In this subsection we give an overview of the evolution in object detection algorithms. We will mainly focus on person detection.

The technique presented in 2001 by Viola and Jones [134] has proven to be a very useful tool to detect faces in natural images. A set of simple features, Haar wavelets, are used to decide whether a human face is present in an image. Haar wavelets compute the pixels under white and black rectangles as illustrated

Figure 4.1: Illustration of a Haar-feature, which is used for face detection. When this feature overlaps with the eye-nose-eye region, it results in a high score due to the fact that the eyes are most often darker than the nose.

in figure 4.1. During a training step in which thousands of face images are presented to the algorithm, an AdaBoosting technique automatically selects which combination of all possible Haar-features is descriptive enough to tell the difference between a face and a non-face. The example in figure 4.1 illustrates the fact that for most faces the eyes are darker than the bridge of the nose. Thus, when this particular feature overlaps with the "eye-nose-eye" region in a face, it results in a high score. Of course, this weak feature alone would not yield a very good face detector, hence it is combined with several other similar features to build a strong classifier combination. A window of 24×24 pixels is slided over the image. In each window 6000 of such features are selected to be calculated and validated. Instead of applying 6000 features in each window, which is time-consuming, the concept of a Cascade of Classifiers is introduced. This method groups the features into different stages of classifiers and applies them one at a time. If a window fails at a certain stage, it is discarded in the following stages. If a window passes all the stages, the algorithm assumes a face is present in that window. In order to cope with different image sizes, each image is downscaled several times. On each scaled image the above-mentioned actions are applied. In 2003 Viola et al. extended and applied their previous work on face detection on the task of pedestrian detection [135], further referred as VJ.

In 2005, Dalal and Triggs [32] proposed a pedestrian detection approach based on the outline of a human silhouette, which is then described by Histograms of

Oriented Gradients (HOG). Their approach works as follows: the orientation of the gradients is stored in histogram bins and weighted with the gradient magnitude. The histograms of an entire evaluated window are fed into a linear SVM. Their approach outperforms the Haar wavelets both in terms of accuracy and processing speed. Even today, many state-of-the-art object detection approaches are in one way or another related to HOG features.

A well-known example is the work of Felzenszwalb et al. [51]. As opposed to the rigid model introduced by Dalal and Triggs, they propose to enrich these rigid models using parts (representing head and limbs when applied to person detection) to increase the detection accuracy in their Deformable Part Model (DPM) approach. In figure 4.2(a) the root model is shown i.e. a standard HOG model in which the outline of a human can be perceived. The additional parts that are calculated at a higher resolution are shown in figure 4.2(b) and the deformation cost for the location of each part relative to the root, is shown in figure 4.2(c). Their approach allows for a slight deviation with respect to the root model. Assume, for example, a rigid HOG model of a human body where the training consists of images where both legs are held next to each other. Applying such a model on an image where a person is walking, as can be seen in the left part of figure 4.2, will most likely fail, due to the difference between the model and the body pose in the image. The DPM approach, on the other hand, detects such a pose using the part models and their allowed deviation. Using this approach, their methods achieved state-of-the art accuracy results on a variety of object classes.

Several attempts were made to improve the detection speed of a DPM-based approach. In 2012, Dubout and Fleuret [44] used a Fourier transform on the HOG features in their Fast Fourier Linear Detector (FFLD). This allows for the calculation of the evaluation of the model as a dot product instead of a convolution, which is indeed less computationally intensive. Using their approach, in which the layers of the feature pyramid are evaluated in parallel, resulted in a reduction of computational cost of factor 7 over the original DPM implementation.

In contrast to adding additional parts to the model, another approach uses additional features (e.g. colour) besides the traditional gradient features. This is done by Dollár et al. [41] in their Integral Channel Features (ICF) approach in which the rigid HOG model is extended with additional channels in which other features are extracted. These channels include six gradient orientation channels, a gradient magnitude channel and three LUV colour channels. Features are extracted as the sum of rectangular regions within the channels. These weak features are used in a decision tree and learned using AdaBoost. In figure 4.3, an example image and the computed channels are shown. The yellow arrows and orange rectangles highlight strong support for the region near the head and

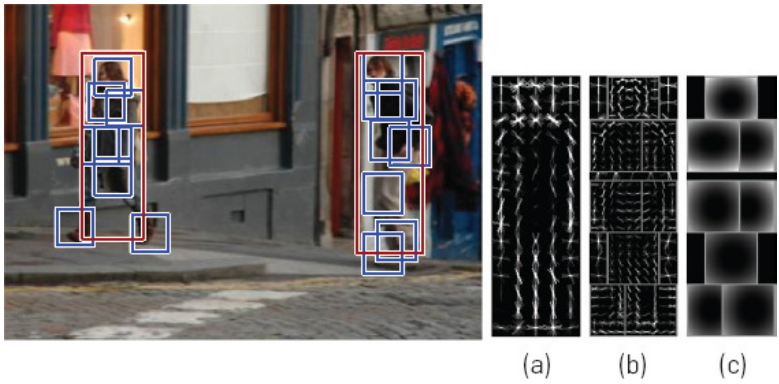


Figure 4.2: (a)Root HOG model, (b) part models representing the limbs and head of a person, (c) The deformation cost for each of the parts with respect to the root model. Image from [51].

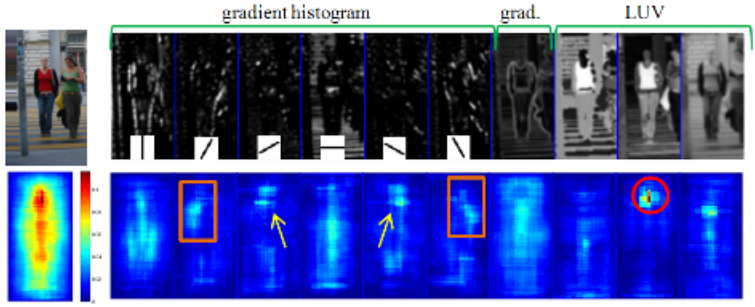


Figure 4.3: Top-row: computed ICF channels on an image patch. Bottom-row: distribution of the selected rectangular features. Image from [41].

shoulders. A surprising but intuitive result is a peak in the support of the 'U' colour channel at the location of the head/face (red circle): apparently, colour is a strong, consistent cue for a person's face/head. In 2010, Dollár et al. [40] proposed the Fastest Pedestrian Detector in the West (FPDW) in which they speed up their detection by not calculating the entire feature pyramid. Other approaches that build on this channel-based detection methodology include the work of Benenson et al. [15], in which high accuracy is achieved by optimising each stage in the detection process.

A final methodology we would like to introduce, includes the use of convolutional neural networks (CNN). CNNs were frequently used in the 1990s, but due to

the rise of the support vector machines they moved out of sight. In 2012, this methodology was again fuelled by Krizhevsky et al. [83] by showing very high image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [114]. Recent applications of CNN include the work of Girshick et al. [58], in which unseen accuracy results were achieved on the PASCAL VOC 2012 dataset. Despite their supreme accuracy, most of the CNN algorithms are far from real-time, making them unsuitable for many applications. However, in recent work of Girshick [57] a Fast R-CNN approach was proposed, in which the required time for both training and detection was significantly reduced.

To summarise this section, we can conclude that regarding object detection, four main approaches are commonly used: Haar-cascade approach, HOG models, DPMs, channel features and CNNs. It is clear we had multiple options to build our automatic person detection analysis tool, so we thoroughly considered both the advantages and the disadvantages of each approach. Since we started in 2013 on our person detection approach, the slow performance of the available CNN approaches made them inapplicable for practical usage. The accurate channel features approach already existed, but these are rigid models. In our application, where we focus on persons in natural settings, various human poses may appear in the images captured by the scene camera. On top of that, we noticed that a person is often not visible from head to toe because of the specific viewing angle of most eye-tracker scene cameras. Therefore, we deliberately opted for a DPM approach for person detection in our application. In order to reduce the computational cost of this approach, we will use the FFLD implementation as proposed in [44]. For face detection, on the other hand, we chose the well-known Viola and Jones method.

4.3 Approach

As mentioned above, in this second part of our approach we focus on the detection of faces and human bodies as a useful methodological step for e.g. customer journey experiments or human-human interaction analysis. In each frame captured by the scene camera we apply a face and person detection algorithm. We then verify whether the gaze cursor overlaps with one of these detections to count how often and for how long the subject looked at another person.

Figure 4.4: Example image in which a person is gesturing and difficult to detect using a rigid model-based approach such as HOG.

Upper body detection

As explained in the previous section, we opted for a DPM-based approach for the detection of human bodies since it allows much more human poses as compared to rigid model approaches. Especially when looking at someone who makes large gestures, e.g. during a presentation as shown in figure 4.4, the deformable aspect of the DPM is crucial. Although the standard DPM person model is widely known as a robust and accurate detector, we noticed some problems when the subject stands close to another person during for example face-to-face communication. As mentioned above, in this case, the person is not entirely visible, making it impossible for even the DPM approach to detect a person in these images.

To overcome this issue, we trained a new human upper body model based on the standard PASCAL VOC 2008 dataset [50]. As a training step we used approximately 8400 positive images (both mirrored and non-mirrored) and about 1000 negative images. Compared to the standard person model, we only used the upper 60% of the labelled bounding boxes of the full human bodies. This percentage was chosen empirically since it corresponds best to the fraction of the human body that is visible in our challenging images. The

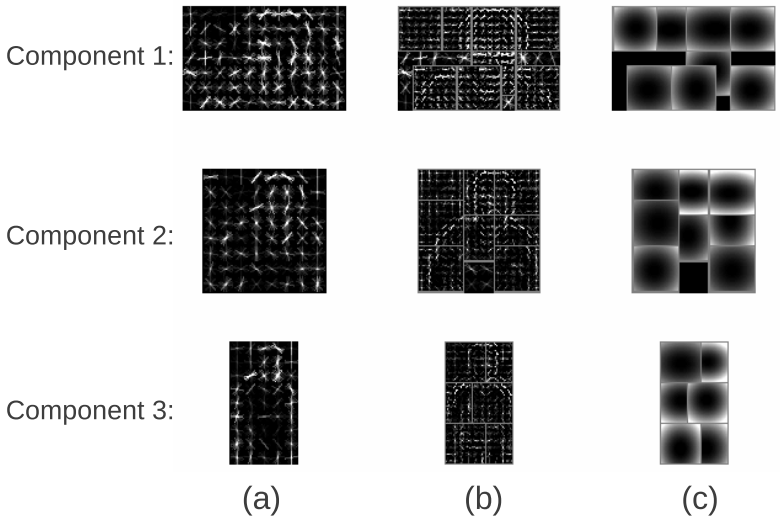


Figure 4.5: Three components of the upper body model, each with their own root model(a), part model(b) and deformation model of the parts(c).

trained detection model consists of 8 parts and three different components ranging from a full upper body model to a model which only consists of the head and shoulders of a person. This way of coping with image border occlusion is also followed by [92], but for a channel features detector. To the best of our knowledge, we were the first to use it on a DPM-detector. An illustration of our upper body model is given in figure 4.5.

We performed experiments to validate the accuracy of our upper body model as compared to full person models. Evidently, we expect a lower performance since less information is embedded into this model, however, such a comparison may reveal a clear and measurable insight into the accuracy. For this validation, we only used the third component since it contained the most information. For verification we compared several person detection algorithms on the INRIA [32] test set. This set contains approximately 300 images in which mostly large scale persons are visible. The results of this validation are shown in figure 4.6.

In this comparison four full person detectors were used besides our upper body model: VJ, HOG, the original cascaded DPM (Latv4-cc-original) and ACF. It is clear that both DPM and ACF perform adequately: as expected their Average Precision (AP) exceeds 80%. In case the traditional 50% overlap criterion for NMS is used for our upper body model, we achieve a proper accuracy (AP=76%) that even outperforms HOG (AP=72%). However, by varying this NMS factor,

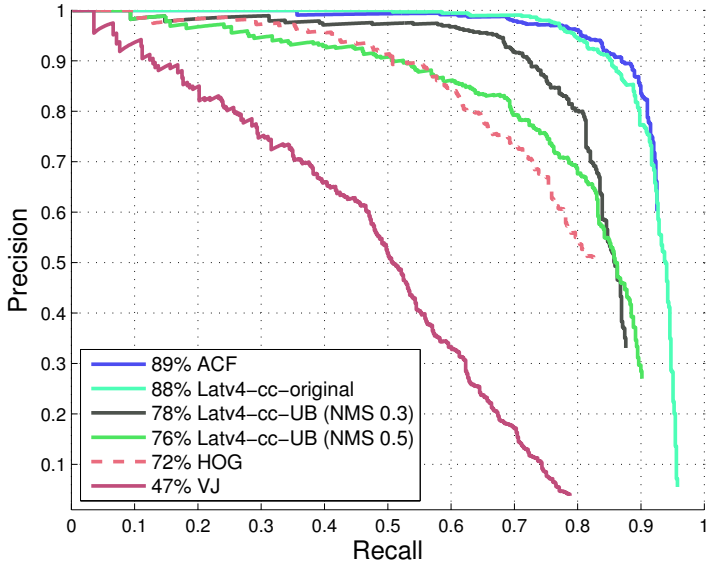


Figure 4.6: Comparing the accuracy of the upper body (UB) detection model against full person detection models on the INRIA test set.

an improvement in accuracy is gained. The optimal NMS value for our upper body detector is found at around 30%. In this case, the detection accuracy increases up to 78%. In our person detection application, we use both the second and third component of our upper body model as shown in figure 4.7.

Next to the successful application of our upper body model in our own application, the newly trained upper body model is already used in other applications as well: for example in the dissertation of Kristof Van Beeck [130], in which our upper body model is successfully applied for the automatic detection of bicyclists in the blind spot zone of a truck.

Face detection

Next to the upper body detection, we also perform a human face detection on each image that is captured by the scene camera of the eye-tracker. For this purpose, we use the OpenCV implementation of the Haar-cascade Viola and Jones [134] face detection. We utilise two types of face models: one for frontal faces and one for profile faces ensuring an accurate detection of the face.

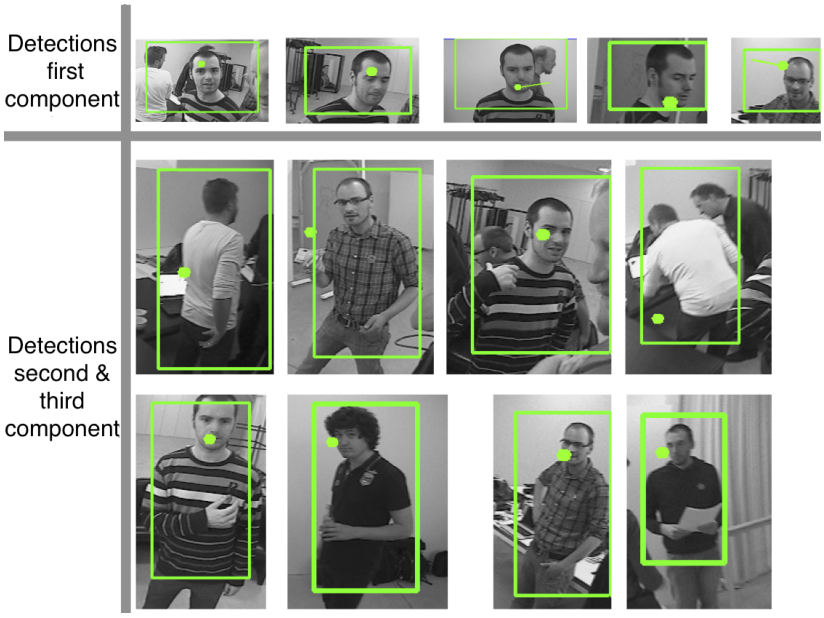


Figure 4.7: Example of the upper body (component 2 and 3 from our model) and head-shoulder (first component from our model) detections.

On top of the traditional Viola and Jones face detection, we use one component of our newly trained upper body model as well for the detection of facial regions in images. When we examine the first component of this model, we can indeed distinguish the contours of a human head and shoulders, as shown in the top row of figure 4.5. Compared to the Haar-cascade model, this DPM model is invariant to various poses of the head. Especially in case of head tilts, which make the eyes (partly) invisible, the Haar-cascade is highly sensitive, whereas the DPM model is highly robust. Example detections of the first component of our upper body model are shown the top row of figure 4.7. In section 4.6, the accuracy of both Haar-cascade and the first component of our model will be validated on actual eye-tracking recordings.

Temporal continuity

Given the upper body and face detection approaches as described above, we are able to automatically analyse eye-tracking recordings by mapping the gaze data on top of these detections. However, some techniques can be used to further

improve the detection accuracy while simultaneously reducing the computational cost.

A first technique includes exploiting the temporal continuity by applying a tracking-by-detection mechanism. Such an approach reduces the computational cost and can be used to remove false detections. This is done using a Kalman filter [73], which is a mathematical filter used to predict the position of both face and upper body. We use a Kalman filter with the following state vector and update matrix, assuming a constant velocity motion model, so that $x_{t+1} = Ax_t$:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where x and y are the position of either the center of the upper body or the face and v_x and v_y are the velocity of respectively upper body or face. The idea behind this Kalman filter is to predict the location of a detection in a next frame, based on the detections in previous frames. Such an approach allows us to define a smaller region in which the upper body and/or face most likely may occur, resulting in a much lower computational cost. Next to reducing the search area, one can also apply the Kalman predictions to fill in missing detections. In case a detection was missing due to occlusion or motion blur, the prediction of the Kalman filter can be utilised. It is needless to say that the amount of successive predictions is limited to a configurable threshold value. Based on our experiments, we allow maximum 5 consecutive predictions. In case no new upper body detection is found within this period, the Kalman filter is automatically disabled until a new detection is found.

The above mentioned approach for the detection of a human upper body performs sufficiently well, but there is still room for improvement. Therefore we propose a second improvement: viz. a temporal smoothing technique (see figure 4.8). Here, we use the gaze data to improve the detection rate, thus minimising both false positives and false negatives. To reduce the number of false positives, we assume that when our system detects that someone is looking at another person, it will last at least a certain time. This minimum duration for example, is 150 ms, which is indeed the minimal fixation duration. However, this duration is tunable via a threshold. This criterion substantially reduces the number of false positives (since many false detections occur short time). On the other hand, if we find short gaps between detection sequences, we can assume those are missing detections. Predicting them will improve the detection rate and thus further reduce the number of false negatives.

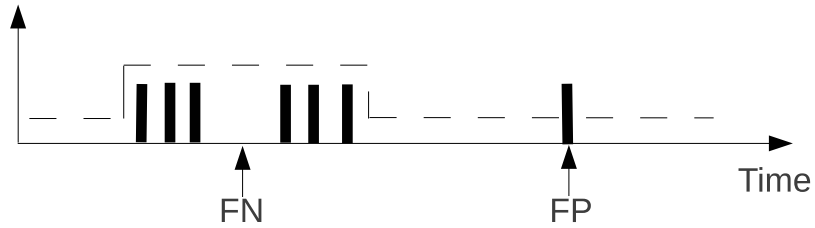


Figure 4.8: Temporal smoothing detection results. Vertical bars: real detections, dashed line: output of the temporal smoothing.

4.4 Semi-automatic analysis

Similar to our object recognition approach we provide a method for manual intervention in case our person detection fails. As mentioned above, we use a tracking-by-detection mechanism in which a Kalman filter predicts missing detections. In case the number of consecutive predictions exceeds a threshold value, for example 5 successive frames, we interrupt the automatic analysis and ask the user to manually annotate the person in the image. This manual annotation is then used as new initialisation of the Kalman filter. Thereafter, the automatic analysis is continued. In case no person was visible in the image, the user can indicate this as well. This will disable the Kalman filter until a new person detection was found. Our initial tests proved that the amount of manual interventions is extremely small. For example, during the analysis of one of our recordings of a presentation which contains 6700 frames, only 15 manual interventions were needed to steer the person detections. It is important to notice that the option for manual interventions can be enabled or disabled depending on the kind of experiment that is analysed. Furthermore, it is possible to parametrise this to balance the accuracy to the amount of manual input.

4.5 Person re-identification

As previously mentioned, mobile eye-trackers are used in the context of human-human interaction analysis. Traditionally, researchers in this field are interested in gaze patterns towards other people (including mutual gaze), which can be calculated using our person and face detection approach. In case a human-human interaction experiment involves multiple participants, like for example a triadic conversation, one would additionally like to know which person the participant is looking at. An illustration of such an experiment can be found in figure 4.9. Here, three persons were equipped with a mobile eye-tracker and

the purpose of the experiment was to investigate the visual behaviour of each participant during a natural conversation. Figure 4.9 is the viewpoint of one of the participants. The red dot illustrates the gaze cursor, which reveals that this participant is looking at the person wearing the yellow sweater in this particular frame.

We expanded our person detection algorithm with a person re-identification step. Such a re-identification allows us to add information on the specific person a participant is gazing at. When we take a look at figure 4.9, it is clear that we could distinguish both persons based on the colour of their clothes. We extract this feature using a histogram comparison as shown in figure 4.10. First, we select a region around each person, as shown in the upper part of figure 4.10. This selection is done manually by drawing a rectangle around each person of interest. For each region, we calculate a histogram, which is a graphical representation of the distribution of pixel values. In this particular example, we calculated two target histograms: one histogram of the person wearing the yellow sweater and another one of the person wearing the black sweater. In a next step, we apply our person detector as explained above. For each frame in which there is overlap between a person detection and the gaze cursor, we calculate a histogram of the detection window. In a last step, we compare the target histograms with the histogram of the detection window, as shown in the bottom part of figure 4.10. Using the highest comparison score, we are able to identify which person the participant was looking at. The results of this approach are thoroughly discussed in chapter 7.

4.6 Results

The validation of our face and upper body detections was done using the same approach as discussed in chapter 3. First, we chose a sequence of 3000 consecutive frames captured using our Arrington GigE-60 mobile eye-tracker during the customer journey experiment in museum M. In this dataset, we manually annotated each person the subject looked at. Thus, when the gaze cursor is positioned close to a person, this person was annotated. We made a distinction between looking at the upper body and looking at the face of the person in the ground truth. Therefore, depending on the position of the gaze cursor, we manually drew a rectangle around either the entire upper body or around the face.

As explained above, our person detection approach automatically draws a rectangle around each detected upper body or facial region in every image captured by the scene camera. Nevertheless, we only consider the detected

Figure 4.9: Sample frame of human-human experiment with three participants. Image from the scene camera of the third participant.

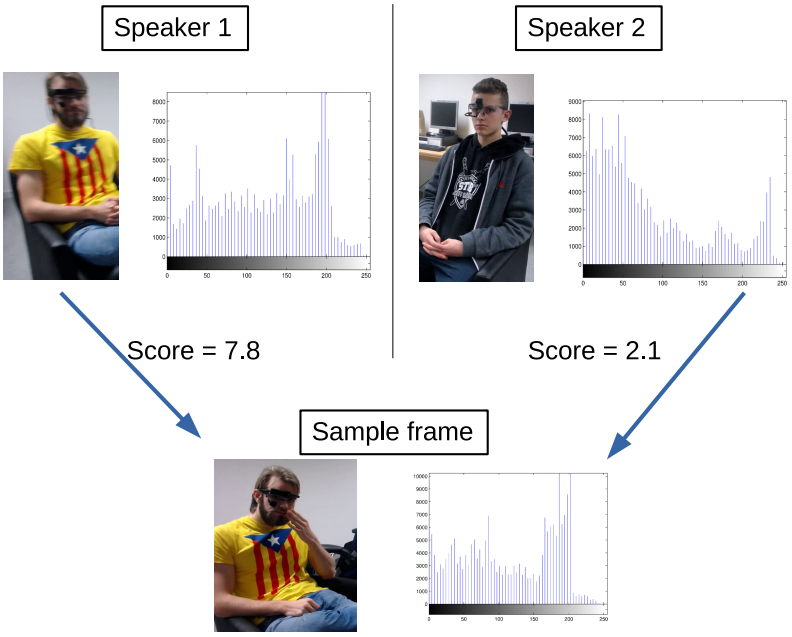


Figure 4.10: Histogram comparison used for person re-identification.

persons whose bounding box overlaps with the gaze cursor. We artificially enlarge each bounding box by a factor 1.10, to ensure sufficient overlap between the bounding box and the gaze cursor.

The validation of our approach was done using the 50% criterion as proposed in [42], which is commonly used to validate object detection algorithms. Here, a detection is considered valid if and only if the bounding box of a detection overlaps at least 50% with the ground truth and vice versa.

In a first experiment, we evaluated the accuracy of our upper body model on real life mobile eye-tracking recordings. In contrast to the validation on the INRIA test set, where only the third component was used, here the second component is used as well. For the validation of this experiment, we only considered the upper body annotations.

In figure 4.11, we present a set of precision-recall curves in which we show the detection accuracy of our system and compare it against a standard full person model. We created these curves by varying a threshold on the detection scores. The blue curve shows the performance of the standard VOC 2009 full body model on our dataset. The green curve shows the performance of the standard VOC 2009 model in combination with our temporal smoothing approach. Indeed, taking into account a minimal fixation length, we managed to improve the detection accuracy. The red curve shows the new upper body model in combination with our temporal smoothing. Mainly in the recall region between 0.8 and 0.9, our approach led to an improvement of 5.53% in average accuracy as compared to the standard VOC 2009 full person model. Given the challenging aspects of the recording we analyse, i.e. changing camera viewpoint, motion blur, changes in illumination etc., we believe that these accuracy results prove the usability of our person detection approach.

In a second experiment we evaluated the accuracy of our face detection approach. In this experiment we only considered the face annotations. First, we evaluated the performance of the standard Haar-cascade approach in which both frontal and profile face models were used. The accuracy of the best parameter setting of this approach is illustrated by the red dot in figure 4.12.

As mentioned in the previous section, the first component of our upper body model can act as an alternative/additional technique to the Haar-cascade face detection. Again, a frame is considered correct if the detection overlaps sufficiently with the respective annotation. For a fair comparison, we ensured sufficient overlap between face annotations and the detections from the first component (head-shoulders) of our upper body model. Therefore, we artificially increased the size of the face annotation bounding boxes using a fixed scaling factor.

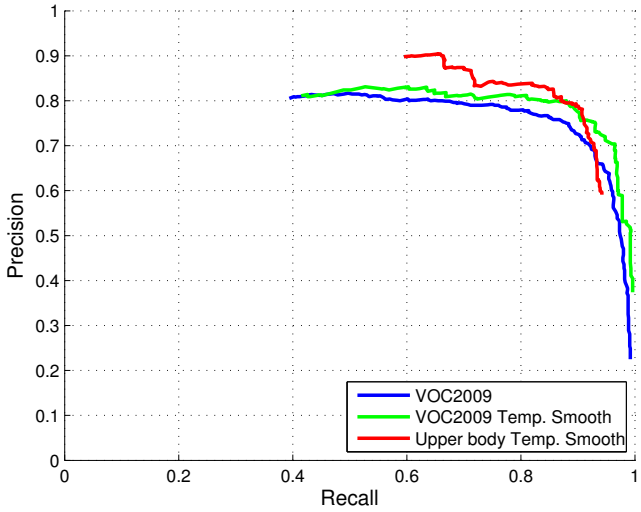


Figure 4.11: Precision-Recall curves of our upper body detection implementation compared to a standard model.

The blue graph represents the performance of our first component. Again we varied a threshold on the detection score to create this curve. This graph reveals a large improvement in accuracy as compared to the Haar-cascade approach. Besides these straightforward evaluations we also tested whether it is valuable to combine both approaches. By applying an OR-operation on the detection results of both Haar-cascade and the first component of our model, we obtain the green curve, which performs significantly better. This idea of combining several detection methods paved the way for the *Combinator approach* as proposed by my colleagues De Smedt and Van Beeck [122].

As shown in table 4.1, we performed a series of experiments to measure the computational cost of our upper body detector. In this table, we tested two images sequences, each containing 1500 consecutive images, that were captured by either the Arrington eye-tracker (images of 320×240 pixels) or the Pupil-Pro eye-tracker (images of 1280×720 pixels). Furthermore, we verified the computational cost of our upper body model for three different implementations: the entire model (which consists of 3 components), a two component implementation (the two last components) and finally an implementation consisting of only the first component. To measure the importance of the tracking mechanism, each test was performed with both tracking enabled as disabled. In case of disabled tracking, the upper body model was applied on the entire image, whereas in case of the enabled tracking,

Table 4.1: Computational cost of our upper body model tested under various parameter settings.

Resolution	**Components**	**Tracking**	**Scale factor**	**FPS**
320×240	1 component	enabled	1	20.83
320×240	1 component	disabled	1	17.04
1280×720	1 component	enabled	1	14.42
1280×720	1 component	disabled	1	2.63
1280×720	1 component	enabled	2	21.13
1280×720	1 component	disabled	2	8.33
320×240	2 components	enabled	1	18.99
320×240	2 components	disabled	1	15.78
1280×720	2 components	enabled	1	14.56
1280×720	2 components	disabled	1	2.31
1280×720	2 components	enabled	2	20.54
1280×720	2 components	disabled	2	7.89
320×240	3 components	enabled	1	12.93
320×240	3 components	disabled	1	7.11
1280×720	3 components	enabled	1	8.11
1280×720	3 components	disabled	1	1.85
1280×720	3 components	enabled	2	12.30
1280×720	3 components	disabled	2	5.15

the upper body model was applied on a cropped region in which the person is probably located. Finally, the images captured by the Pupil-Pro eye-tracker were scaled down by factor 2 (new resolution is then 640×320 pixels), since in that case the person was still detectable by the upper body model.

This table reveals the importance of the tracking mechanism, in particular on the 1280×720 pixel images. Furthermore, we notice that the performance of the model in which only 1 component or 2 components are evaluated, is indeed quite high. Especially in case of the 640×320 pixel and the 320×240 pixel images, about 20 frames per second are processed. The significant increase in computational cost of the 3 components evaluation is due to the presence of the mirrored version of each component.

Besides these small-scale experiments, we used our person detection approach for the analysis of several challenging and long-lasting eye-tracking experiments as will be described in chapter 7.

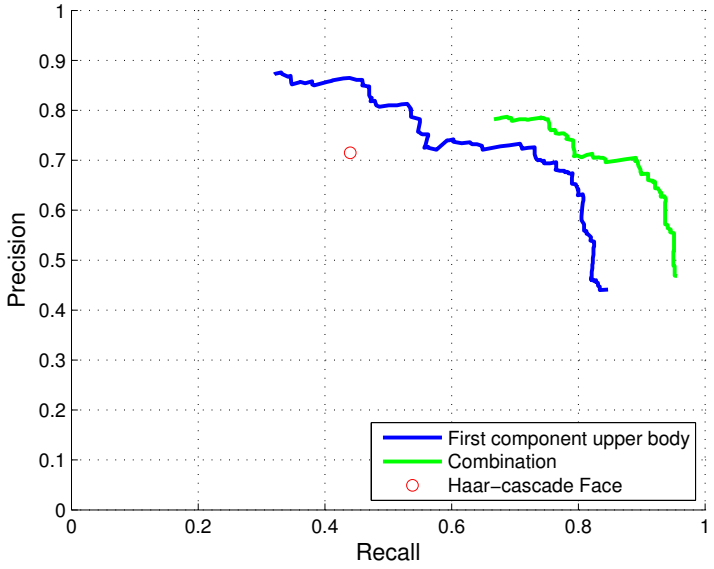


Figure 4.12: Precision-Recall curves of face detection compared to upper body detections.

4.7 Conclusion

In this chapter we presented an approach for the automatic analysis of eye-tracker data based on both face and person detection. Our approach is suited for counting how often and for how long one looked at a person or a face. On top of that, we developed a person re-identification approach that provides information about which person in particular the subject was looking at. In order to further improve the detection rate, we proposed two novelties. The first is a temporal smoothing approach based on the gaze cursor to avoid many false positives and false negatives detections. Secondly, we trained a new DPM model which is designed for upper body detections and which can be used as alternative/additional face detection as well. The applicability of the newly trained torso model was proven using an analysis of an eye-tracker experiment which was conducted in a museum context. Finally, we developed a method for manual intervention in case our person detector fails. Using such a methodology ensures highly accurate analysis of mobile eye-tracking experiments that is significantly less time-consuming as compared to fully manual analysis.

Chapter 5

Hand detection

Since mobile eye-tracking found its way into various research domains in which human interaction is studied, there is a growing interest in the visual behaviour towards body parts that are instrumental to communication, such as faces, (upper) bodies and hands. The approach we presented in chapter 4 allows us to (partly) automate the analysis of the visual behaviour towards faces and human bodies. To this date, the analysis of recordings in terms of how often and how long the subject looked at the hands or the gestures of another person is done manually. To address this issue, we present the third part of our analysis framework, i.e. a semi-automatic hand detection approach. Here, we search for human hands in images, captured by the scene camera of a mobile eye-tracker, and we map the gaze data on top of these detections to gain further insights into visual attention towards relevant articulators in face-to-face communication.

This chapter is subdivided into five main parts. In section 5.1, we introduce mobile eye-tracking applications in which the automatic detection of human hands is relevant. Section 5.2 gives an overview of existing hand detection approaches. In section 5.3 we explain the importance of manual interventions in this part of our analysis framework. In section 5.4 we discuss the hand detection approach that we developed. Finally, in section 5.5, we validate accuracy and speed of both our approaches using real life mobile eye-tracking recordings.

Our first hand detection approach was presented at the VISAPP 2015 conference [37] and as a chapter in *Computer Vision, Imaging and Computer Graphics Theory and Applications* [39]. The second hand detection approach was presented at the VISAPP 2016 conference [38].

5.1 Introduction

Our motivation for developing a highly accurate hand detector comes from the growing applicability of mobile eye-tracking in a variety of disciplines including computer science, linguistics, sociology and psychology. Common in these research fields is the interest in visual behaviour towards relevant body parts. Besides looking at faces and human bodies, one is often interested in the visual attention towards the hands as instruments of transaction and main articulators of communicative gestures. An example application in human-human interaction is shown in figure 5.1. Here, the subject, wearing a mobile eye-tracker, receives an object from someone else. In these give-and-take experiments, researchers are particularly interested in the visual distribution between looking at the face of the other person and looking at the hand in which the object is held. In this example frame, the subject is currently looking at the right hand of the other person, as shown by the red dot, representing the gaze cursor. Another application is research on the visual behaviour towards gestures [22, 67]. Mobile eye-tracking is, among others, used to gain insights into the influence of (pointing) gestures on visual behaviour. An example question to be answered in such an experiment is: Does a subject shift his or her visual attention towards the pointing direction of a co-participant?

Since we aim to perform the analysis of real-life and unobtrusive mobile eye-tracking experiments, in which participants can move freely, it is obvious that a marker-based approach is inapplicable. Again, currently the analysis of this type of recordings is done manually, which requires substantial annotation work [22]. Furthermore, this time-consuming task restricts the efficient creation of annotated recordings, which are vital for new research.

As already mentioned in the previous chapters, the eye-tracking community would greatly benefit from the implementation of techniques that reduce the manual annotation load. Therefore, we propose a technique to (semi-) automatically detect hands in video data recorded by the scene camera of a mobile eye-tracker. By mapping eye gaze data on interlocutors' body parts that are instrumental to face-to-face communication (like hands and faces), a first step in the analytical process is realised, as it allows for basic calculations of visual distribution. These data can then serve as the basis for further analytical work (e.g. the analysis of visual fixations on certain gesture types), which is thoroughly discussed in chapter 6.

Detection of human hands in real-life images is an extremely challenging task due to their varying shape, orientation and position. In our application on the other hand, the complexity even increases since a) hands appear relatively small in images (in some recordings a human hand is less than 20 pixels wide) b) due

Figure 5.1: Illustration of human-human interaction. The red dot represents the current visual focus of the subject wearing the mobile eye-tracker.

to the natural interactions captured in our recordings, the hands often move fast, which introduces motion blur (e.g. during gesturing).

Recently, several highly accurate hand detection algorithms were developed for 3D images. Hand detection in 2D images, however, is a far from trivial task due the lack of depth context. Several attempts were made, including skin-based detection, model-based detection or pose estimation techniques. Unfortunately, when applied to real-life images their performance drops significantly.

On top of the challenging task we try to tackle, we aim to develop a generic method to achieve a high detection rate. It is well known that fully automatic approaches typically do not guarantee high accuracy in practical cases. To overcome this, we again introduced an intelligent mechanism which automatically asks for manual input when the confidence of a detection is below a threshold value. Using such an approach increases the detection rate significantly, at the cost of a limited number of manual interventions.

In the course of this PhD project, we developed two hand detection approaches. The first method implies a 3-stage approach to generate an optimal result. First, the search space is reduced, using an upper body detector. Second, we make a hypothesis using a sliding window approach of a hand model combined with a skin-based hand detection. Third, to ensure reliable detections, we use a tracker and an advanced elimination approach to remove false detections.

The second method abandons the use of a computational intensive hand model. Instead, skin colour segmentation is applied in combination with an intelligent tracking mechanism of multiple entities. Furthermore, a novel validation of the entire upper body is performed to ensure the correctness of the body parts. As mentioned above, a method for manual intervention is introduced in each approach.

Furthermore, during our study, we noticed that it is hard to find fully annotated video material of human hands in real life recordings. Therefore we made our annotated dataset of eye-tracker recordings publicly available as described in [37, 39]. This dataset contains three sequences in which approximately 4400 human hands were manually annotated[1].

5.2 Related work

In this section we give an overview of existing hand detection techniques, which can be divided into four main categories: coloured gloves, motion sensors, depth information and hand detection in traditional 2D images. Furthermore, we discuss the limitations of these approaches.

A first well-known method for hand detection is the use of coloured gloves, which are used as a marker that can be easily detected in images. In [136] Wang and Popović use a multi-coloured glove, enabling the detection of various hand orientations and poses. Since we focus on hand detection in natural and unconstrained scenes, we cannot afford the use of coloured gloves, since they have a profound influence on the visual attention during a conversation.

A second approach of hand detection makes use of motion sensors [124]. Typically, multiple sensors like ultrasonic transmitters and inertial sensor modules are placed on the arms and hands of the user. Because of the same reason as mentioned above, it is not recommended to place additional sensors on the participants due to possible interference with the natural behaviour.

The increasing popularity and public availability of 3D cameras paved the way for a third type of hand detection as for example the approach of Zhou et al. [110]. These cameras provide useful depth information of a scene. Depth information facilitates hand detection and it even enables the detection of small items such as for example fingertips [107]. Although this is a promising approach, it is not applicable in our application since most of the egocentric cameras, and in particular mobile eye-trackers, are not equipped with 3D sensors.

[1] `http://www.eavise.be/insightout/Datasets/`

A last approach of hand detection is based on image processing in 2D images without the need of additional markers or sensors placed on the body. A first method for finding a human hand in an image is done using skin segmentation in various ways. One may use manual fine-tuning of a set of sliders selecting a threshold on specific colour channels such as proposed in [56, 116]. Often the images are therefore converted to another colour range such as HSV, since skin segmentation in RGB is known to be sensitive to slight illumination changes. Other approaches automatically detect a face [134] in an image and use this for the extraction of skin information [98]. In [70] a statistical colour model was developed, allowing the calculation of skin-tone probability of each pixel.

Skin segmentation is often combined with monitoring the velocity of the skin regions. Obtaining this motion is done in several ways: a basic approach is to calculate the displacement in subsequent frames, but more sophisticated methods such as Mixtures of Gaussian (MoG) are also widely applied [143]. In [89], an approach for modelling non-verbal communication was presented and here a 2D hand likelihood map was developed. This map follows the assumption that in an image, the hands are skin coloured and that they show more movement than the face, which is obviously also skin coloured. In [10], the same assumption was followed and here the motion of the skin regions was applied to further refine the hand detections.

Other methods for the detection of human hands exist as well, either or not combined with the skin segmentation. In [80] a hand tracking approach was described based on Kanade–Lucas–Tomasi (KLT) features in combination with colour cues. Such an approach yields good results as long as the hand is easily visible (i.e. large enough) in order to calculate an adequate number of features. This approach is not applicable in our type of experiments, where the hands only represent a small part of the image, as can be seen in figure 5.1. In [120] a real-time hand tracking method is presented using a mean-shift embedded particle filter. Their system is very fast (only 28ms per frame is needed) but the resolution of their test images is only 240×180 pixels. In their experiments they only detect and track a single hand, whereas in our application we need to track and disambiguate both hands with respect to the human pose.

Bo et al. [18] present a hand detection technique based on a combination of Haar-like features and skin segmentation. This approach is sufficiently accurate in controlled scenes, e.g. a clean white background, but the approach suffers from high false positive rates when applied to less constrained scenes. Another accurate approach was proposed by Mittal et al. [98], combining a deformable part model (DPM) of a human hand with skin segmentation to generate hand candidates. Those candidates are then suppressed using a super-pixel-based non-maximum suppression yielding accurate detections. This technique has a large computational cost due to the complexity of the DPM and the calculation

a super-pixel representation of the entire image, yielding an average processing time for a single frame of 1280×720 of about 290 seconds. Another model-based approach is found in [76] and is capable of locating parts of interest in a robust and precise manner, even when the surrounding context is highly variable and deformable. Applied to hand detection, the chains model generates a feature chain between an easily detectable object, such as a face, and the object of interest (i.e. the hands). The work of Spruyt et al. [123] is also a recent hand detection approach focussing on real-time Human Computer Interaction (HCI). However, compared to the images we tackle, the difficulty of the datasets they used is limited, in that they do not involve typical challenges of real-life data, like e.g. changing camera angles and distances, (partial) occlusions of and by hands, etc. These are situations in which their approach fails.

Many hand detection approaches work well for still images, however when applying them to recordings where persons move naturally, a problem arises. Since these algorithms obtain no information regarding the human pose, it is impossible to discriminate left and right hand. Nevertheless, such a distinction is indispensable in gesture analysis. In [89], next to the hands, a face is also detected, making the hand positions relative w.r.t. the position of the person. This is combined with a synthetic 3D polygonal torso model, resulting in an approximated 3D pose of the upper body of the person. Another approach for distinguishing left and right hand is found in [16]: here, next to applying a model for the detection of a human hand, they also use specific models for left and right hand, allowing them to differentiate both hands. In the work of Eicher et al. [47], a technique for estimating the spatial layout of humans in still images is presented. They use a combination of upper body detection and the detection of individual body parts. This method performs well on larger body parts (such as arms or heads), whereas smaller parts (e.g. hands) are much more challenging. The accuracy of this technique largely depends on the upper body detection. Detection at a wrong scale will result in deviating limb detections. Furthermore, their approach works far from real-time: an average of 25 seconds is needed to process a single 1280×720 frame. A similar approach was proposed by Yang and Ramanan [138]. They present a method for human pose estimation in static images based on a representation of part models, in which they take into account the relative locations of parts with respect to their parents (e.g. elbow w.r.t. to shoulder), resulting in accurate detections. However, the authors admit their approach has difficulties with some body poses e.g. raised or fully stretched arms.

Based on a comparison of the previously described techniques, we opted for the work of Mittal et al. [98] as a starting point for our first approach. This method achieves decent accuracy and its source code is publicly available, which allows for easy comparison. In the subsection 5.4.1 we discuss the

modifications we made in order to improve the detection results and reduced the computational cost significantly. Furthermore, we explain the integration of manual interventions in case our hand detector would fail.

Our second approach, which is discussed in subsection 5.4.2, differs significantly from all previously mentioned ones. We propose a hand detection methodology, which is both fast and accurate, and which allows for manual intervention. We extensively optimised and combined previously described techniques, and integrated them with probabilistic information.

5.3 Semi-automatic analysis

It is important to highlight that we tackle an annotation application that to this day is typically done completely manually. It is well known that repetitive tasks, such as a frame-by-frame inspection, are error prone. Therefore, to ensure correctness, a cross-validation over multiple annotators is often mandatory for the analysis. As a result, such an analysis is expensive in terms of man-hours.

Our goal is to reduce the amount of manual analysis as much as possible while in the meantime reducing the analysis time without compromising the accuracy, which is often a contradictory demand.

As discussed in the previous chapter, our automatic analysis of visual behaviour towards faces and upper bodies is highly accurate, which makes the amount of manual interventions negligible. The detection of human hands, on the other hand, is far more complex. Even the best approaches fail to reach top accuracy on realistic recordings. To overcome this burden, we developed a system that automatically detects hands in images and calculates a confidence measure of each hand, based on multiple cues. When the confidence of a hand drops below a user-defined threshold, our automatic analysis is paused and the user is asked to manually annotate the corresponding hand, using a user-friendly GUI. After this intervention, the automatic analysis is resumed. We fine-tuned the parameters of our system to ensure the lowest amount of manual interventions possible, while guaranteeing high accuracy.

The importance of automating this detection step of the analysis is unmistakable. Not only is the manual annotation labour reduced to the minimum, our semi-automatic approach turns the repetitive task into a task in which sporadically manual input is requested, making it less error prone. Therefore, it is no longer mandatory that multiple annotators spend time on the same recording. Furthermore, the ability of manual interventions ensures a certain level of

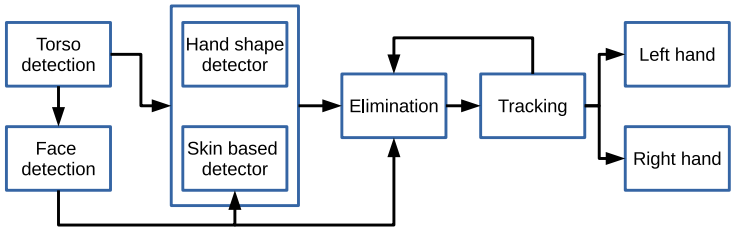


Figure 5.2: Graphical representation of the model-based hand detection approach. The three stages: upper body and face detection, hand detection and a combination of elimination and tracking.

control, whereas fully automatic approaches are often black-box systems in which interpretation and/or correction of false detections is much more complicated.

Since the confidence calculation differs between our approaches, we present the exact implementation in the respective subsections.

5.4 Approaches

As already mentioned in section 5.2, we developed two different hand detection approaches. In this section both approaches are discussed. First, we propose our model-based hand detection approach and then, based on the limitations of this method, we introduce our segmentation-based hand detection approach.

5.4.1 Model-based approach

An overview of our first hand detection approach is given in figure 5.2. The general idea is that we first detect a human upper body in the image, yielding a robust reference for the detection of smaller body parts. Furthermore, the face of the person is detected as well. After that, we detect hands using a model introduced by Mittal et al. [98] in combination with a skin-based detection. Then we apply an advanced elimination scheme to remove false detections. Finally, we use a Kalman filter to track left and right hand using the spatial relationship of consecutive frames. In the remainder of this dissertation, this approach is referred to as model-based hand detection.

Upper body Detection

The first stage in this approach is the detection of a human upper body, for which we use our self-trained upper body model as presented in the previous chapter. Using this model, rather than the more widely used full person detector, has the advantage that we can cope with images in which a person is not completely visible (from head to toe).

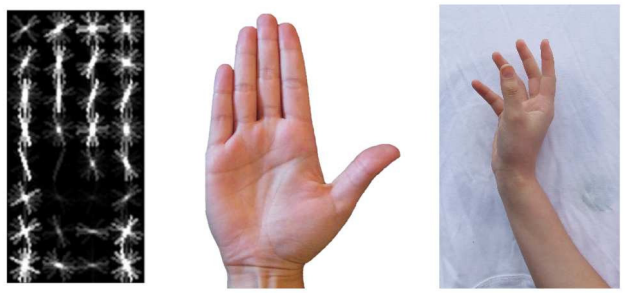
Face Detection

The next stage is a face detection step. Again, we use the same methodology as proposed in the previous chapter. Information retrieved from the face detection is used to further improve the accuracy of the hand detections. In the work of Mittal et al. [98], the face detection is only used for improving their detection results by applying skin segmentation. The colour of the face is then used to segment the entire image. In our approach, on the other hand, we make use of the proportions of the face to reject hand detections which have an abnormal size compared to the size of the face. This is based on the general rule that a human face has more or less the same size as an outstretched human hand. We do allow some deviations to the size of the face in order to cope with some depth variations such as when a hand points towards the camera and therefore appears larger than normal in the images.

Hand Detection

Once our system identifies the presence of a person using our upper body detection, we run our actual hand detection algorithm. Instead of searching for hands in the entire image, we define a search area by expanding the upper body detection bounding box in both vertical and horizontal orientation. By doing so, we reduce the area in which we search for hands. As mentioned before, we started from the work of Mittal et al. [98]. This implies that we use the same DPM-model of a hand, as illustrated in the left part of figure 5.3. In their approach, an additional context model is used, as illustrated in the rightmost part of this figure. However, the experiments we ran for this study showed that the addition of this model introduces a significant amount of false detections, as a result of which we opted not to use it.

The hand model was developed to detect vertically oriented hands, but in real-life recordings any hand orientation is possible. Therefore, we rotate the enlarged region around the detected upper body, and apply the hand model on each rotated image, yielding an accurate detection of hands in any orientation.



(a) Illustration of the hand models. The left image is the HOG representation of the hand model. The middle image illustrates the hand model, while the right image is an illustration of the context model (hand and its surrounding region including the background and wrist).

(b) Example frames that were used by Mittal et al. [98] for training the hand and context model.

Figure 5.3: Illustration of the hand model(a), and an illustration of some sample images that were used for training the hand and context models(b).

Table 5.1: Accuracy of the hand model versus rotation angle of the images.

Step size	**Precision**	**Recall**	**Time/frame**
10 °	79,20 %	78,86 %	42 s
20 °	75,78 %	75,47 %	21 s
30 °	71,24 %	71,13 %	14 s
45 °	62,82 %	62,55 %	9,3 s
90 °	48,72 %	48,50 %	5 s

An illustration of this rotation is given is figure 5.4, in which two different step sizes are shown. In table 5.1, various step sizes are applied on a set of 100 annotated frames of 1280×720 pixels. As expected, by applying a larger step size, the processing is faster, but the accuracy drops significantly. Since we target a post-processing application, in which achieving high accuracy is more important than processing time, we opted for a step size of 10° per rotation. To decrease the computational cost related to this type of model evaluation, we used the Fourier-based acceleration approach of Dubout and Fleuret [44], as already introduced in the previous chapter.

The accuracy performance of the hand model is sufficient, as long as a hand is clearly visible in the image. However, when a hand is not visible or strongly deformed — for example due to motion blur caused by fast movements of the arms — this model shows low detection rates.

To overcome this problem, we developed an additional hand detection technique as shown in figure 5.5. This technique segments the image in skin and non-skin based on three different colour spaces as introduced by Rahman et al. [17]. In this work, skin colour is defined in both Red Green Blue (RGB), Hue Saturation Value (HSV) and Luma Chroma blue Chroma red (YCbCr) colour spaces, resulting in a robust detection mechanism for various skin tones, even under different lighting conditions. An overview of the segmentation rules as developed in [17] is given in equations 5.1, 5.2 and 5.3. These segmentation rules are then combined into a final formula as shown in equation 5.4. When a pixel meets S, it is segmented as skin, otherwise as non-skin. We opted for this segmentation approach since a) the authors validated that their combination of colour spaces outperforms traditional approaches and b) the total segmentation can be performed very fast since the segmentation of the individual colour spaces is easily divided over multiple threads.

$$\begin{cases} r_1 = (R > 95) \wedge (G > 40) \wedge (B > 20)) \\ r_2 = (max\{R, G, B\} - min\{R, G, B\} > 15) \\ r_3 = (|R - G|) \wedge (|R - G| > 15) \wedge (R > G) \wedge (R > B) \\ r_4 = (R > 220) \wedge (G > 210) \wedge (B > 170) \\ r_5 = \wedge(|R - G| \leqslant 15) \wedge (R > B) \wedge (G > B) \end{cases} \tag{5.1}$$

$$\begin{cases} c_1 = Cr \leqslant 1.5862 \cdot Cb + 20 \\ c_2 = Cr \geqslant 0.3448 \cdot Cb + 76.2069 \\ c_3 = Cr \geqslant -4.5652 \cdot Cb + 234.5652 \\ c_4 = Cr \leqslant -1.15 \cdot Cb + 301.75 \\ c_5 = Cr \leqslant -2.2857 \cdot Cb + 432.85 \end{cases} \tag{5.2}$$

$$\begin{cases} h_1 = H < 25 \\ h_2 = H > 230 \end{cases} \tag{5.3}$$

$$\begin{aligned} rule1 &= (r1 \wedge r2 \wedge r3) \vee (r4 \wedge r5) \\ rule2 &= (c1 \wedge c2 \wedge c3 \wedge c4 \wedge c5) \\ rule3 &= (h1 \vee h2) \\ S &= rule1 \wedge rule2 \wedge rule3 \end{aligned} \tag{5.4}$$

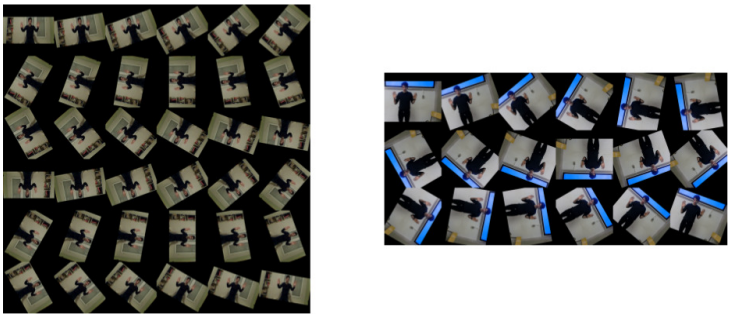


Figure 5.4: Illustration of the rotation of our images in order to detect hands in any orientation. Left: step size is 10° per rotation. Right: step size is 20° per rotation.

Since we no longer depend on the accuracy of the face detector for skin segmentation, our approach generates far more hand candidates as compared to the work of Mittal et al. [98]. We apply this segmentation to the enlarged upper body detection as shown in figure 5.5(b). Next, we skeletonise this result using a sequence of several erosion and dilation steps in order to get an accurate estimation of the skeleton, as illustrated in figure 5.5(c). In a following step, we apply the information obtained from the face detector. We use the correlation between the human body parts to classify the skeletonised image.

If a skeletonised part has a length that is similar to the height of the face, we classify it as a hand, as illustrated by the top row in figure 5.5. Parts that are larger than a face are automatically treated as an arm, as illustrated by the bottom row in figure 5.5. For each part that is classified as an arm, we estimate a hand at both endpoints of that arm, as illustrated in figure 5.5(d). Purple boxes illustrate the hand classifications, blue boxes the arm detections and green boxes the estimated hands at the endpoints of the arm. Estimated candidate detections at the wrong endpoints are rejected using the elimination and tracking described in the next sections.

Elimination

After the above-mentioned steps, a large amount of hand detections is obtained, as seen in figure 5.6(a). The task of this elimination stage is to reject non-hand detections and to cluster overlapping detections. The output of this elimination operation is a reduced number of hand candidates as shown in figure 5.6(b). In our elimination process we apply the following steps:

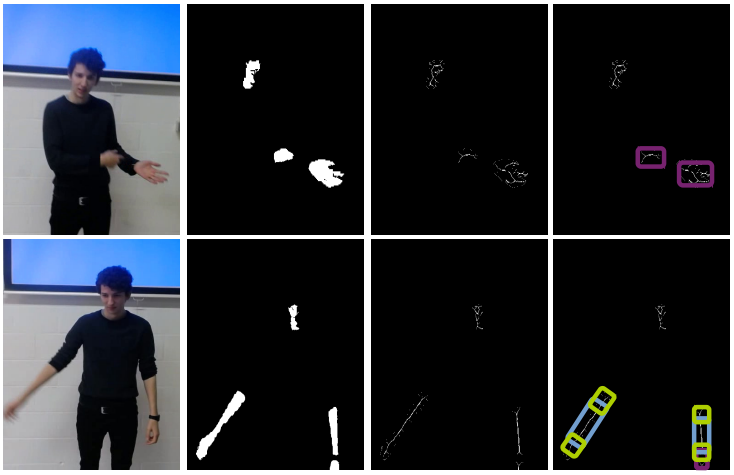


Figure 5.5: From left to right: original image(a); binary image based on skin segmentation(b); skeletonization(c); arm and hand estimation(d).

- Remove hand detections (that are obtained by the model evaluation) which have an insufficient number of skin pixels, using the same skin detection algorithm as described in the previous step.
- Remove hand detections (that are obtained by the model evaluation) which have a divergent size with respect to the size of the face.
- Cluster overlapping detections and hand candidates based on their overlap and distance between their centres. Scores of the clustered detections are aggregated into a single score per cluster.
- Reduce the contribution of clusters that coincide with the face. We noticed that a face is often detected by the hand model. Eliminating these detections is not a viable option since persons can hold their hands in front of the face. Therefore, we reduce the score of those overlapping clusters by a predefined factor to minimise the impact.

In the elimination step, we reduced the number of hand detections. Finally, we classify the remaining clusters in a final detection for left and right hand using a Kalman filter as explained in the next section.

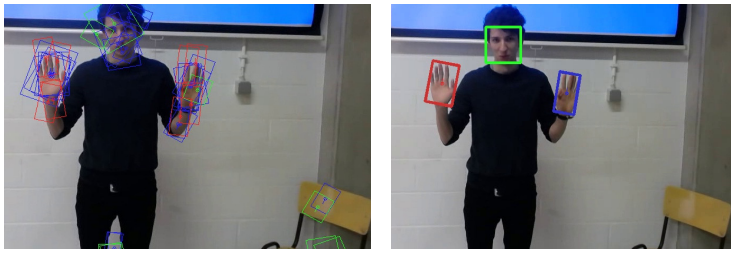


Figure 5.6: Left: large amount of detections before elimination; Right: final clusters after elimination step.

Tracking

Our tracking stage, which is of vital importance to achieve maximal accuracy, is realised by exploiting the spatio-temporal relationship between consecutive frames. Therefore, a Kalman filter [73] is used. Similarly to the one proposed in our person detection approach, we use a constant velocity motion model, so that $x_{t+1} = Ax_t$. This mathematical filter is used to predict the position of the hands, which is needed when no hand is found due to e.g. occlusions. A second advantage of using a Kalman filter is that the noise on the measured position of the final detections is filtered out, resulting in more stable detections. For each detected upper body in an image, two additional Kalman filters are defined: one for the left hand and one for the right hand in order to track each hand individually.

For each of the remaining clusters, as described in the previous section, we calculate the cost, based on the distance, to assign them to one of the Kalman filters. By choosing the cluster with the lowest cost, we select the best candidate for each Kalman filter. Using this approach ensures that only two clusters remain: one for the left hand and one for the right hand.

To summarise this section, we give an overview of our contributions as compared to the approach of Mittal et al. [98]:

- Reduced computational footprint of our algorithm by avoiding both super-pixel calculation and the validation of the context model.
- Reduced search space by using an upper body detector and only searching for hands in a region around the upper body detection. This resulted in a lower computational cost and it reduced the number of false detections.

Furthermore, by defining a relationship between an upper body and hands, we avoid searching for hands in images where no person is visible.

- Skin-based detection is performed even when no face is detected, resulting in more detection candidates.
- Elimination of false detections using the size of the face.
- Kalman tracker for both left and right hand.

Semi-automatic analysis

As mentioned in section 5.3, we integrate a method for manual intervention in our automatic hand detection approach. For each detected hand a confidence score is calculated, based on several cues. The key idea is that when the confidence drops under a specific (user-defined) threshold, our algorithm requests manual input from the user, who then has to manually annotate the missing hand(s).

Relying only on the detection score results in a significant amount of manual interventions. To overcome this, we also take into account the distance between the chosen cluster and the predicted position (coming from the Kalman trackers). Our formula of the confidence score M is shown in equation 5.5:

$$M = \alpha \log(D_{max} - D) + \beta S_i \tag{5.5}$$

where:

$$D = \begin{cases} D_{max} - 1, & \text{if } d(C_i, C_{i-1}) \geq D_{max} \\ d(C_i, C_{i-1}), & \text{otherwise} \end{cases}$$

D_{max} stands for the maximum allowed distance between the current chosen cluster detection and the final cluster in the previous frame, C_i and C_{i-1} define respectively the centre of the current and the previous cluster. S_i stands for the cluster score, while α and β are used to change the weight of the distance and detection score. In our experiments, we empirically determined the optimal value of those parameters: $\alpha = 0.5$ and $\beta = 1.0$. This confidence measure is calculated for both hands.

The general concept of this approach is that a detection is likely to be valid if either the distance to the predicted location (based on previous detections) is low or if the detection score is high. If M of a hand drops below a user-defined threshold, manual input is requested. After this manual intervention, the state vector of the corresponding Kalman filter is reset, resulting in a stable reference point for further detections. By varying this threshold, we can change the amount of manual interventions from zero (fully automatic detection) up to the number necessary to achieve full accuracy ((semi-)automatic detection).

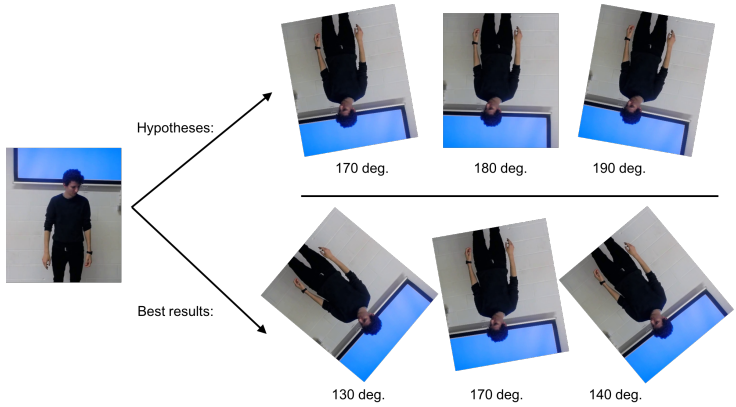


Figure 5.7: Illustration of our reduced orientation concept. Top part: hypothesis of orientations that would result in the best detection scores. Bottom part: actual rotations that obtained the best detection scores.

Compared to the work of Mittal et al. [98], our approach is more accurate and we even reduced the computational cost as discussed in section 5.5. Unfortunately, since we still need to evaluate the hand model on 36 rotated versions of each image, our approach remains slow. To lower this computational cost, we performed some initial experiments in which we tracked the orientation of each hand as well. This would allow us to reduce the number of orientations in which we perform the evaluation of the hand model.

An illustration of this reduced rotation is shown in the top part of figure 5.7. Suppose that, based on previous frames, we know that the hands are facing downwards in a particular image, then our hypothesis is that when we rotate the image 180 degrees (since the hand model was trained on vertically oriented hands) and we allow a slight deviation by evaluating rotation angles of 170 degrees and 190 degrees as well, our system should be able to detect the correct hands in a subsequent image.

This hypothesis was tested in an additional experiment. Here, we applied the hand model to a set of 491 images (further referred to as D2) without additional skin segmentation nor tracking. Each image was rotated 36 times in order to detect hands in each orientation. We restricted the scale space to the desired dimension, since all hands appear approximately at the same size in this dataset. Furthermore, an additional NMS step was applied to cluster overlapping detections that are found in multiple orientations. The result of this initial experiment is shown in the blue graph in figure 5.8. It is clear that the accuracy performance of only the hand model is rather limited, however it

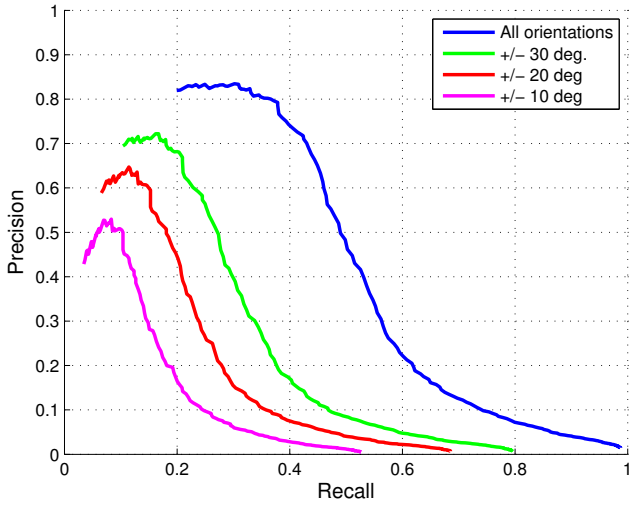


Figure 5.8: Accuracy of hand model that is applied on a limited number of rotated images. Blue curve represents the accuracy of all orientations. Other curves represent the reduced orientations.

serves as a starting point for validating our hypothesis.

In a next step, we reduced the rotation angles that were used for evaluation in order to reduce the computational cost. For this purpose, we annotated the orientation of each hand in this dataset. Based on these annotations we reduced the orientations at which the hand model is evaluated. This reduction was done in three experiments. In the first experiment, we allowed 30° of deviation in each direction. Thus, for a hand that was annotated at 60°, the hand model is evaluated on rotated versions of this image in the range of 30° up to 90°. The accuracy of this experiment is illustrated by the green PR-curve in figure 5.8. The same experiment was repeated for both 20° and 10° deviation in each direction as shown by the red and magenta curves in figure 5.8.

This experiment reveals that restricting the number of orientations by which each image is rotated results in a significant drop in accuracy. It is clear that hands are often found at unexpected rotation angles as shown in the bottom part of figure 5.7, in which we illustrate the top three orientations that resulted in the best detection scores for the given image.

Based on this experiment we conclude that the descriptiveness of the hand model is inadequate to estimate the hand orientation. This is most likely caused by

the extremely large range of different appearances of the object over which the model is not able to generalise well. A human hand is indeed a very articulated object (more than 20 DOF), with a wide range of possible poses as clearly shown in the example images from the training set that was used for developing this hand model as shown in figure 5.3.

As a result, the concept of reducing the number of orientations is inapplicable. To further reduce the computational cost, the next subsection presents a new segmentation-based approach, in which the computationally intensive model is no longer used. Nevertheless, the focus of this new method remains the development of a highly accurate hand detection approach, in which the amount of manual interventions is as low as possible.

5.4.2 Segmentation-based approach

As illustrated in figure 5.9, our segmentation-based approach is a combination of several processing blocks, but it does not rely on Mittal's hand model. A first step is the detection of a human upper body, which is used to identify the presence of a person and to reduce the search area. This step is inherited from the model-based approach. Next, we apply a skin colour segmentation, which is used to generate hand candidates. To further enhance the detections, a multi-entity tracker is used for temporal smoothness. Finally, we validate the hand candidates using a) a comparison between a predicted position and the candidate and b) a validation of the upper body pose. Each step of this workflow will be discussed below.

In contrast to our previous approach, which can run fully automatic, this segmentation-based approach requires manual intervention in the first frame of each recording. Once the first frame is manually annotated, our method uses this starting point for further automatic processing. Again, for each hand a confidence score is calculated. Once this score drops below a user defined threshold, the automatic analysis is paused and manual input is asked from the user.

Context retrieval

The first stage of our approach is identical to the model-based hand detection approach: i.e. human upper body and face detection. The upper body detection is used to detect the presence of a person in the images and is also used to reduce the search area for the hands: we extended the width of the upper body detection by a factor 3.5 and the height by a factor 1.8. These factors

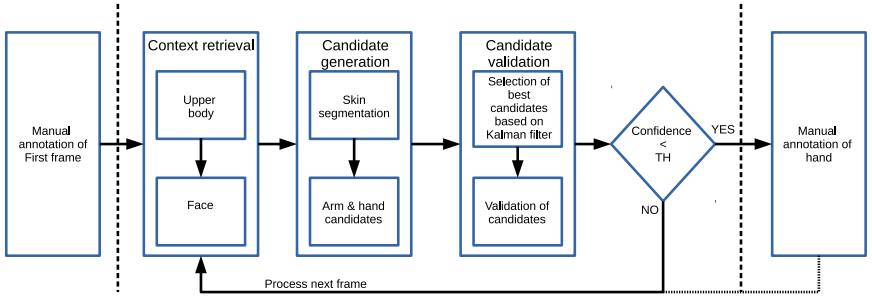


Figure 5.9: Workflow of our segmentation-based hand detection approach.

are determined empirically and ensure that any possible hand position of a person lies within the extended region of interest. This step allows us to restrict searching for hands within this region and to discard the rest of the image. In Figure 5.10(a) the original upper body detection is displayed using the purple rectangle, while the blue rectangle illustrates the extended bounding box. The pink rectangle illustrates the face detection, which is similar to the one used in the model-based approach.

Candidate generation

We segment the image patch, which is the extended bounding box, in skin and non-skin using the same segmentation rules as in our model-based approach. However, we added an additional segmentation rule in which the skin colour as obtained by the face detection is taken into account as well. This idea is based on the work of Dollár et al. [41], in which they prove that the 'U' channel in the LUV colour space is a strong and consistent cue for detecting a person's face. For each detected face, we calculate the median 'U' value of the center of the detection. This $\widetilde{U}$ is then taken into account in the skin segmentation, as shown in equation 5.6. Parameters (α=8) and (β=2) allow a slight aberration to $\widetilde{U}$. Our experiments revealed that equation 5.6 even outperforms the segmentation rule that is applied on the H colour channel in the HSV colour space as shown in equation 5.3. Therefore, when a face is detected, the segmentation rule as shown in equation 5.6 is used instead of equation 5.3.

$$\begin{cases} u_1 = U > \widetilde{U} - \alpha \\ u_2 = U < \widetilde{U} + \beta \end{cases} \tag{5.6}$$

Figure 5.10: Generation of hand candidates: a) original image, b) skin segmentation, c) contour detection, d) fit ellipse, e) final hand candidates.

An illustration of this segmentation is given in figure 5.10(b). After two dilation and two erosion steps, we fit a contour over each group of pixels that is sufficiently large, as shown in figure 5.10(c). This criterion is derived from the size of the face as already introduced in the previous approach. We keep track of the size of the face to overcome missing face detections. In a next step, a bounding ellipse is fitted over each contour (figure 5.10d). Each endpoint of the major axis of an ellipse is treated as a possible hand candidate, as illustrated by the green dots in figure 5.10e. The example shown in figure 5.10(d) contains four ellipses. One coincides with the face and is therefore automatically discarded, another one is found on an approximately skin-coloured chair and two ellipses overlap with the arms. In total, the major axes of three ellipses remain in this example, hence six possible hand candidates are found.

Candidate validation

The final stage of our approach is developed to automatically select the best candidate for both left and right hand and to validate them. The temporal continuity of the image sequence is exploited using a Kalman filter, which is similar to the ones used in our model-based approach. The selection of the best candidate for both left and right hand is done by choosing the hand candidate with the smallest distance to the Kalman prediction of the respective hand. As mentioned above, each hand candidate belongs to a line (i.e. the major axis of the ellipse). Therefore, when an endpoint is chosen as the best candidate for a hand, the remaining endpoint of that line can be seen as a joint. In case the person is wearing long sleeves, this joint corresponds to the wrist. On the other hand, when a person wears short sleeves, the joint corresponds to the elbow.

Since we obtained the location of the upper body and the face, we are able to estimate the location of both left and right shoulder. This estimation is based on the height of the face and the width of the upper body bounding box. Examples of these rough skeleton representations are given in figure 5.11. Compared to the model-based approach, in which a detection consists of a bounding box

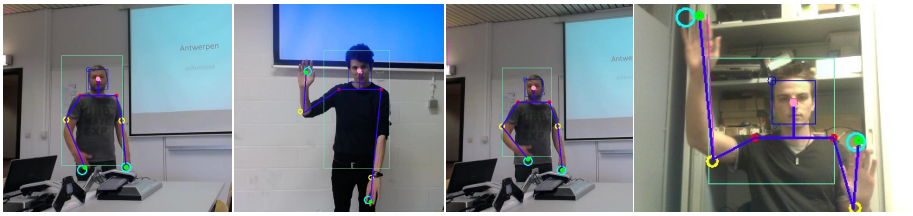


Figure 5.11: Examples of our detections on the four datasets. Green circles are the hand detections, yellow circles are the corresponding joints, red circles indicate the estimated shoulder positions.

Figure 5.12: Example frames from the Buffy dataset [52] indicating the large variety of human poses within this set. From this labelled dataset, our probability maps (P_{Elbow}) and (P_{Wrist}) are derived.

around a hand, our segmentation-based approach was built to detect and track a single point that indicates the extremal point of each hand. In the last image of this figure an example is given in which the left joint corresponds to the wrist.

In a final step, our approach automatically calculates the probability that the obtained skeleton representation is correct. In particular, the relative position between the shoulders and the respective joints is validated. We hypothesise that when the endpoint of the joint (elbow or wrist) is correct, the other endpoint a.k.a. the hand, is likewise correct.

We built probability maps to create a map of possible and valid positions of elbows and wrists w.r.t. the shoulder. These maps are created using the original labelling of the publicly available *Buffy dataset* [52], more specifically we used the labels of wrist, elbow and shoulder. The motivation to use this particular dataset comes from the large variety of human (arm) poses that are recorded in this dataset, as can be seen in the sample frames in figure 5.12.

For each image in this dataset, we calculate the relative position of elbow and

wrist w.r.t. the shoulder, resulting in four sets, each containing 1496 data points. In figure 5.13 the data points for both left elbow and wrist are shown. Two mirrored sets of points are used for the right shoulder. Next, we apply a Gaussian smoothing resulting in a dense map, which is then normalised. In total, four probability maps were developed: elbow w.r.t. shoulder (P_{Elbow}) and wrist w.r.t. shoulder (P_{Wrist}), each for both left and right side.

Finally, we weighted -in each frame of a recording- the relative position between an estimated shoulder position and the respective joint to these probability maps. Each candidate position is weighted to both (P_{Elbow}) and (P_{Wrist}) since it is unknown in advance whether the joint corresponds to either a wrist or an elbow. Thus, for each joint, two probability scores are calculated. In case both scores drop below a certain threshold, we assume that the joint position is invalid. Then, since both hand end point and joint are connected (i.e. by the major axis of the ellipse), we assume that the hand position is invalid as well. In that case, our automatic analysis is paused and we ask the user for manual intervention as described in the next paragraph. For clarification, figure 5.14 illustrates situations in which the joint position was invalid and thereby manual intervention was requested. It is clear that in each example where the position of the joint is invalid, the hand position is likewise wrong.

Semi-automatic analysis

Similarly to our model-based approach, we provide a method to manually intervene, in case the automatic analysis fails. Again, our goal is to reduce the amount of manual interventions as much as possible, however without sacrificing accuracy levels. As mentioned above, the weighting result from the probability maps is taken into account in the confidence calculations. Besides this cue, we also use the distance D (i.e. the distance between a final hand position and the respective Kalman prediction) and the number of consecutive hand predictions $pred$ that are used (thus no valid detection was available). The calculation of the confidence condition C is given below:

$$\begin{aligned} C = \{(D > D_{max}) \wedge (pred > pred_{max})\} \vee \\ \{(max(P_{Elbow}, P_{Wrist}) < P_{TH})\} \end{aligned} \tag{5.7}$$

D_{max} stands for the maximum allowable distance between prediction and hand candidate. This maximum distance depends on the size of the person and is therefore calculated as follows: 0.75×face width, $pred_{max}$ stands for the maximum amount of predictions that is allowed. Finally, P_{TH} stands for the lowest probability value that is allowable. If condition C (equation 5.7) is met,

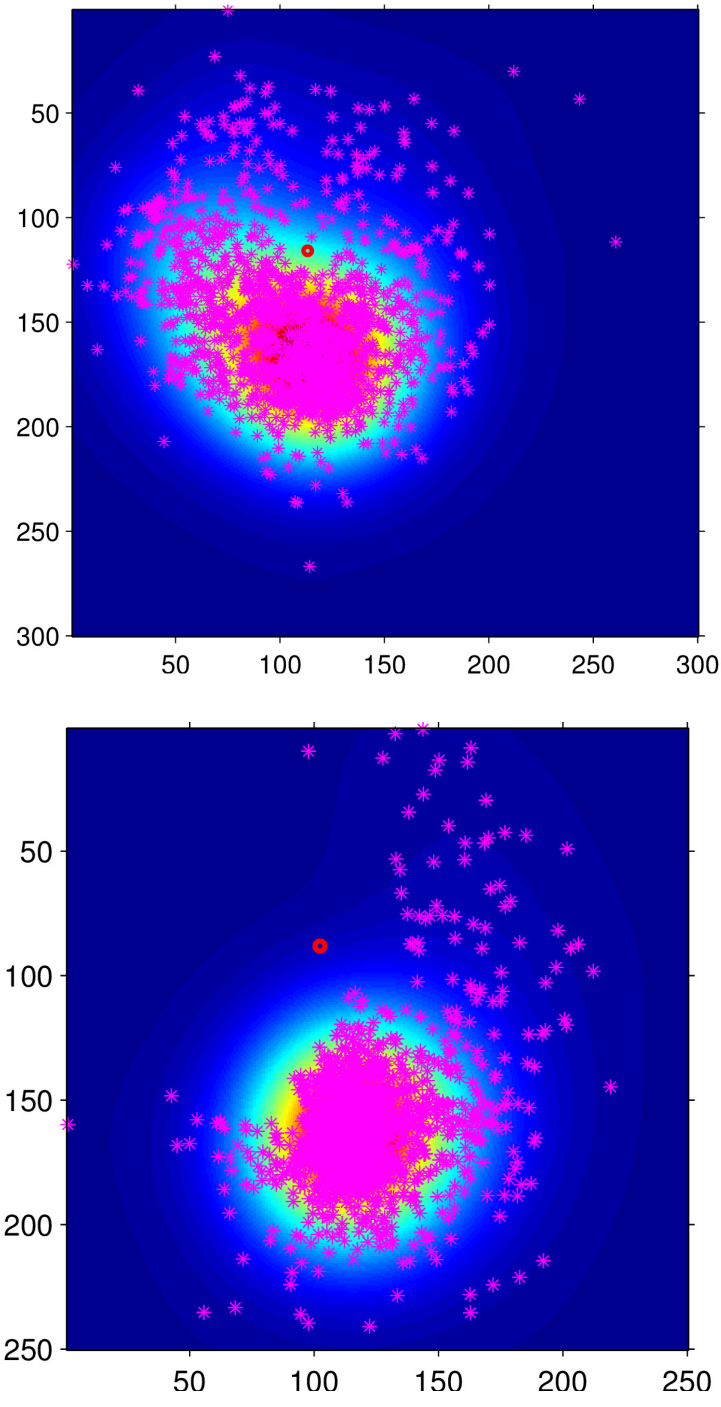
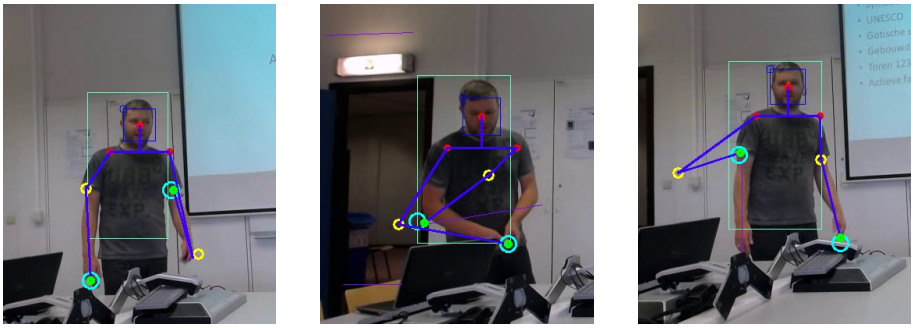


Figure 5.13: Top image shows data points and probability map of the left wrist w.r.t. left shoulder(P_{Wrist}). Bottom image shows the data points and probability map of left elbow w.r.t. left shoulder(P_{Elbow}). The red dot in each map illustrates the position of the left shoulder.



(a) (b) (c)

Figure 5.14: Examples in which the shoulder-joint position was wrong. (a) left arm: joint and hand are swapped (b) left arm: both joint and hand are wrong (c) right arm: position of joint is completely wrong.

our system automatically pauses and asks for manual intervention, as described above. Otherwise, the next image is processed automatically.

By varying the above-mentioned parameters, one can increase or decrease the amount of manual interventions. It is clear that in case the strictness of the confidence is lowered, our system requires less manual interventions, but this obviously comes at a cost of lower accuracy.

We implemented an additional feature in our approach to reduce the amount of manual interventions. As mentioned before, the probability maps are developed using the data labels from the Buffy dataset. Although this dataset contains a large variety of human poses, it may occur that a particular pose of a wrist or an elbow corresponds to a low probability since this particular pose occurs only sporadically in the Buffy dataset. When the automatic processing is paused due to an insufficient probability score, the user can indicate that the particular joint position is nevertheless correct. In that case, the probability map is updated making this joint position valid in future processing.

A video of our segmentation-based approach is available online[2].

[2] `http://youtu.be/DsxdBc4gGjg`

5.5 Results

In this section, we evaluate the accuracy of our hand detection approaches. Similarly to the previous chapters, our focus lies on the accuracy of the developed approaches (i.e. capability of detecting hands in images), whereas in chapter 7 the hand detection approaches were used for the actual analysis of large-scale eye-tracking recordings. In that chapter, we do evaluate the link with the gaze data to count how often and for how long the subject was looking at hands during a human-human interaction experiment.

First, we introduce the datasets that were used, next we discuss the accuracy of our two approaches compared to other techniques. Furthermore, we explore the contribution of the manual interventions to the accuracy performance. Finally, the computational cost of both approaches is evaluated as well.

5.5.1 Datasets

During our research we noticed it was challenging to find video material with fully annotated hand positions in consecutive frames. In [98] a dataset of annotated movie frames is presented, but unfortunately, the available frames are not consecutive, making them unsuitable for our approach which exploits the spatio-temporal relationship between consecutive frames. We also examined video recordings from the MPI archive[3], which were annotated in terms of gestures but contain no additional information of hand locations.

We did find two publicly available datasets that we can partially use for this purpose. The first one was introduced in [123] and contains artificial recordings of a person gesturing in front of a webcam. For validation, we used a recording containing 1251 video frames in which the location of both hands was annotated. This dataset is further referred to as D4. The second dataset is the '5-signers' dataset [24], which contains time-series data of the hand positions collected from 5 signers during performance of sign language. Each of the signer sequences contains 39 frames resulting in 390 annotated hand-instances. In the remainder of this section, this dataset is referred to as D5. An illustration of this dataset is given in the rightmost image of figure 5.15.

To overcome the lack of fully annotated video material, we created some recordings ourselves using a Pupil-Pro mobile eye-tracker. In the first recording two persons stood face-to-face at a distance of 3 meters from each other. The subject, wearing the eye-tracker, was told to look attentively at the interlocutor, who made fast and large movements with his hands and arms. From this rather

[3] `http://corpus1.mpi.nl`

Figure 5.15: Illustration of each of the datasets used for the validation of our hand detection approach.

artificial recording we chose a short sequence of 403 consecutive frames in which both left and right hand were manually annotated. This recording is further referred to as D1. The second and third recordings were performed in a more natural setting. In this experiment, a presentation was attended by two subjects. We chose a sequence of one subject in which in 491 consecutive frames the positions of both hands of the presenter were manually annotated. This set is further referred to as D2. Finally, for the third sequence we annotated the first 1300 frames of the recording of the other subject, which is further referred to as D3. This third sequence is an extremely difficult set for hand-tracking since the hands are often occluded by furniture. We specifically included this set since, because of its challenging nature, it fully exploits our algorithm and reveals its shortcomings. An illustration of these recordings is given in the first three images of figure 5.15. Since it is hard to find publicly available hand-annotated video material, we made our reference dataset publicly available[4] allowing other researchers to benchmark their algorithms on these recordings.

In total, we have 4388 annotated hand instances in our own recorded datasets and 2892 annotated hand instances from the publicly available datasets. This results in 7280 annotated hand instances that can be used for validation.

5.5.2 Accuracy model-based approach

Since we are interested in expressing the accuracy of our hand detection approaches w.r.t. the amount of manual interventions that were needed, we opted for another accuracy measure rather than the traditionally used PR-curves or ROC-curves. We chose the F_1 -measure, which is the harmonic mean of precision and recall as shown in equation 5.8. In our validation on the other hand, where a) in each frame used for validation only one person is visible and two hands are always annotated, and b) our hand detection approaches always return two hand detections for each upper body detection, the number of false

[4] http://www.eavise.be/insightout/Datasets/

Table 5.2: F_1-measure of our model-based approach both with and without tracking and compared against other hand detection approaches. In this experiment the option for manual interventions was disabled.

	Mittal [98]	Yang [138]	model-based	model-based incl. tracking
D1	85.0%	24.2%	**83.4%**	**88.2%**
D2	48.9%	46.5%	**52.9%**	**65.3%**
D5	77.6%	n.a.	**81.1%**	**n.a.**

positives (FP) and false negatives (FN) are equal, hence precision equals recall. Therefore, F_1 equals both precision and recall.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5.8}$$

A hand detection is considered valid if the distance between the detection and the manual annotation was below half-face width, which is a commonly used measure for hand detection algorithms [143]. We compare our results to the performance of two state-of-the-art hand detection techniques. The publicly available hand detection algorithm of Mittal et al. [98] was chosen, in which we use the two best scoring detection as candidates for left and right hand. We also compare to the pose estimation proposal of Yang and Ramanan [138], in which we classify the outermost bounding boxes of the arms as hands.

First, we tested our model-based hand detection algorithm (section 5.4.1) without tracking of the hands nor manual intervention to provide a fair comparison with the existing fully automatic approaches [98, 138]. We performed this experiment on the first two sequences of our own dataset (D1,D2) and the '5-signers' dataset (D5). The result of this initial experiment is shown in the third column *(model-based)* of table 5.2. It is clear that our method, even without tracking, outperforms the other approaches. Although a note on the bad performance of the approach of Yang et al. [138] should be made. The detection code we have used, was developed to detect poses of persons from head to toe, whereas in the images of D1, the legs of the person are not completely visible as shown in the first image of figure 5.15. By enabling the tracking, we further improved the accuracy as shown in the rightmost column of table 5.2. No tracking information was available for recording D5, since the 39 frames of each sequence were not consecutive.

In a second experiment, we enabled the manual interventions of our model-based approach to examine the improvement in accuracy. Figure 5.16 plots the accuracy (F_1-measure) relative to the amount of manual interventions expressed

as a percentage of the detectable hands in the respective sequences. This experiment was done on our own recordings (D1,D2 and D3).

By assuming that manual intervention is always correct, it is obvious that 100% of manual intervention results in an accuracy of 100%. In this experiment, we varied the threshold that is applied to score M from equation 5.5, to lower the amount of manual interventions. It is clear that our approach remains highly accurate even when the amount of manual interventions is heavily reduced.

Furthermore, these graphs show a significant improvement in accuracy by allowing a minimum amount of manual interventions as compared to fully automatic analysis. In case of recording D1, the accuracy increases from 88.2% to 93% at the cost of only 7 manual interventions out of the 403 frames. On D2, on the other hand, the accuracy improves with even 12% at the cost of only 14 manual interventions out of the 491 frames.

The accuracy results of D3 are somehow surprising. As already mentioned, recording D3 is much more challenging as compared to the other recordings since often the hands are (partially) occluded. Furthermore, the hands were not completely visible in the first frames of this recording, which caused wrong initial detections that were mistakenly tracked during the entire recording. However, this reveals the full potential of our approach since a limited amount of manual interventions (7.7%) leads to a tremendous improvement in accuracy (94.9%).

5.5.3 Accuracy segmentation-based approach

As described in section 5.4.2, our segmentation-based hand detection approach was developed as a less computationally intensive alternative for the model-based approach. Nevertheless, the main focus remained developing a highly accurate hand-detection algorithm in which a minimum amount of manual interventions is necessary. The validation of our segmentation-based approach was done on our own recorded datasets (D1,D2 and D3) as well as on D4. In total the annotated positions of 6893 hands were available for validation. Table 5.3 gives an overview of these validation experiments. It is important to mention that we empirically determined the optimal values of ($pred_{max} = 5$) and ($P_{TH} = 5$), which were used the calculation of C in equation 5.7.

The leftmost columns of table 5.3 show the accuracy and corresponding amount of manual intervention of our semi-automatic approach. In this initial experiment, the validation using the probability maps was disabled. It is already clear that this approach is highly accurate, at the cost of a minimum amount of manual interventions. In the middle columns, the same experiment was repeated, however, here the validation based on the probability maps was enabled. As

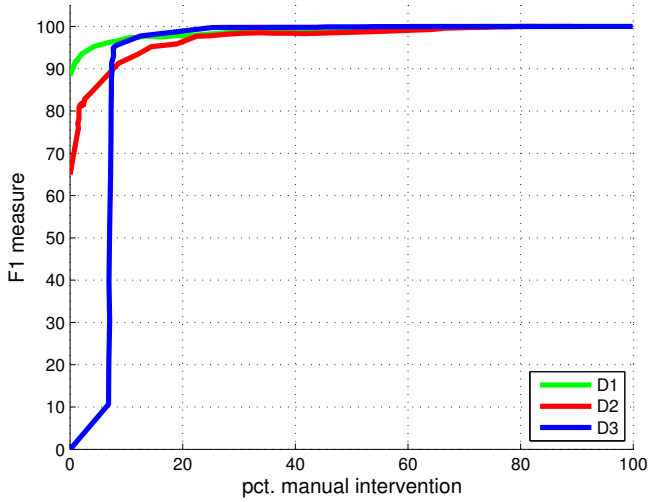


Figure 5.16: Result of our (semi-)automatic approach in which accuracy is improved by manual interventions.

Table 5.3: Comparison of our segmentation-based hand detection approach and our model-based approach. *man.* indicates the amount of hands that were manually annotated.

	segmentation-based without prob		segmentation-based with prob.		model-based	
	man.	F-measure	man.	F-measure	man.	F-measure
D1	1.63%	90.85%	**2.62%**	**95.76%**	4.2%	95.28%
D2	2.55%	83.57%	**1.84%**	**92.75%**	19%	92.13%
D3	0.65%	81.08%	**0.75%**	**88.31%**	8.6%	87.62%
D4	n.a.	n.a.	**2.47%**	**97.89%**	n.a.	n.a.
avg.	1.61%	85.17%	**1.92%**	**93.68%**	6.72%	91.4%

expected, on average a little more manual interventions were requested, but the accuracy increased significantly.

In the rightmost third column of table 5.3, we show the performance of our model-based approach as tested on datasets D1, D2 and D3. To allow for a fair comparison, we show the amount of manual interventions that is required in the model-based approach to achieve a similar accuracy as achieved by our segmentation-based approach. As seen, the amount of manual analysis is substantially higher as compared to the segmentation-based approach.

One may suggest that combining both the model-based approach and the segmentation-based approach would yield an even lower amount of manual interventions. Indeed, instead of directly asking for manual intervention when condition C is met, we could apply the hand model to locate the hands in these images and use the detections to steer the Kalman filters and thereby avoid manual interventions. However, we hypothesise that such an approach will fail, since in the majority of the cases in which manual intervention was required, hands are indeed hard to detect.

To validate this hypothesis, we performed an additional experiment, in which we applied the hand model on the images of D2, in which manual intervention was requested. Again, each image was rotated 36 times in order to detect hands in any orientation. The accuracy of this experiment is shown in figure 5.17 by a precision-recall curve, in which we varied a threshold value on the detection scores. It is clear that these images are indeed challenging. The accuracy performance on this subset of images is significantly less than the average performance on the entire dataset as validated in section 5.4.1 in figure 5.8. This experiment reveals that the additional use of the hand model is not useful in these circumstances.

Evidently, the accuracy of our proposed segmentation-based approach is sensitive to the clothing of a person. When, for example, more skin is visible than expected by our approach, our proposed method remains applicable. However, one can expect that more manual interventions are required than normal. Although we aim to analyse real-life experiments, we can assume that somehow the clothing of the participants can be controlled.

5.5.4 Computational time

We also compared the execution speed of our approaches, as shown in table 5.4. It is clear that the execution time of our model-based approach algorithm is drastically lower compared to the other pre-existing techniques on the same hardware (Intel Xeon E5645). Our approach is much faster compared to the work of Mittal et al. [98], since we no longer depend on the super-pixel calculation. We also outperform the computational cost of Yang and Ramanan [138] by a factor of 3. Despite our efforts, the model-based approach remains slow for practical use. Our segmentation-based approach on the other hand is about 240×faster as compared to our model-based approach. Our first approach needed approximately 36 sec for processing an image of 1280×720 pixels as shown in table 5.4, whereas our new approach only requires 150 ms to process the same frame. This significant improvement in computational speed is mainly achieved by abandoning the computationally intensive DPM-model.

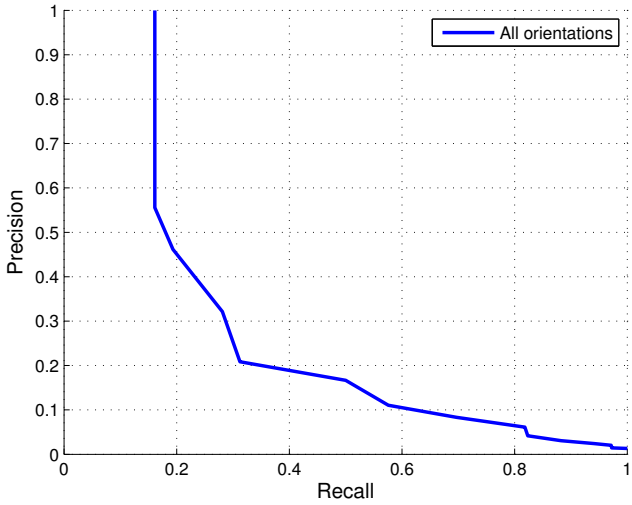


Figure 5.17: Result of hand model applied on images where manual intervention was requested.

Table 5.4: Execution times per frame averaged over all frames of 1280×720 pixels.

	Avg time/frame
Mittal [98]	293.33 s
Yang [138]	113 s
model-based approach	**36.67 s**
segmentation-based approach	**150 ms**

5.6 Conclusion

In this chapter, we presented two approaches for the detection of human hands in challenging real-life image sequences. Our approaches include a novel semi-automatic way of improving the accuracy, i.e. a generic mechanism that finds the optimal moments to ask for manual intervention, resulting in a much higher accuracy with minimal manual effort. By calculating a confidence score for each hand, based on multiple cues, we measure the reliability of each detection. By thresholding this value, we can adapt the number of manual interventions.

Our first approach is built on the work of Mittal et al. [98]. We extended this approach in order to improve the accuracy and to lower the computational

cost. We reported good accuracy as compared to state-of-the-art techniques, while the computational cost is significantly lower. Despite our efforts, the computational cost remained too large for practical usage. To overcome this problem, we proposed our next hand detection approach.

The second approach no longer uses the computationally intensive DMP-model. Instead a segmentation-based approach is proposed, in which a novel validation of the human upper body pose ensures accurate hand detections. We validated our approach using several datasets and compared them against our model-based approach. This validation reveals that our approach is more accurate than the model-based approach while being more than 240×faster, which makes it more applicable in real-life applications. Moreover, our system requires an even lower amount of manual interventions in order to achieve the same accuracy.

In chapter 7, our segmentation-based approach is used for the analysis of real-life mobile eye-tracking recordings. Furthermore, in that study we compare both efficiency and accuracy of such an analysis against traditional manual analysis.

Chapter 6

Gesture detection

In a variety of research fields, including linguistics, psychology, sociology and behavioural studies, there is a growing interest in the role of gestural behaviour related to speech, gaze and other modalities. The analysis of multimodal communication requires high-quality video data and detailed annotation of the different semiotic resources under scrutiny. In the majority of cases, the annotation of hand position, hand motion, gesture type, gesture position, etc. is done manually, which is a time-consuming enterprise requiring multiple annotators and substantial resources. Building on our semi-automatic hand detection approach as presented in the previous chapter, we present the fourth part of our analysis framework: i.e. a gesture analysis approach.

The remainder of this chapter is structured as follows: in section 6.1, we motivate the need for an automatic gesture analysis approach within the above-mentioned research fields. In section 6.2, we give an overview of existing gesture analysis approaches and we highlight their shortcomings. Section 6.3 discusses our entire gesture analysis approach. In section 6.4 a profound validation of the accuracy performance of our approach is given.

Our gesture detection approach was submitted for review in the journal *Language Recources and Evaluation* [34].

6.1 Introduction

The increase of customer available mobile eye-tracking devices sparked the interest in visual behaviour during natural communication. More specifically,

researchers are interested in visual behaviour towards body parts that are relevant in communicative settings. As described in the previous chapters, our proposed framework enables the (semi-)automatic analysis of mobile eye-tracking recordings. By mapping the gaze data on top of face, upper body or hand detections, we get a first, rough, insight into visual behaviour.

Another vital aspect of research on human communication and interaction, in which mobile eye-tracking is often used, focuses on the interplay between speech and gesture in construing and coordinating meaning. Such multimodal recordings would allow for research on *multimodal patterning*, i.e. the study of recurrent co-occurrences between verbal and nonverbal resources (e.g. markers of obviousness co-occurring with shoulder shrugs, hesitation markers such as *uhm* co-occurring with gaze aversion, etc., see [53] for an overview).

One of the main challenges for this type of analysis, independent of the specific research question or approach, is gaining access to qualitatively and quantitatively rich data. Experimental and corpus-based studies rely on high-quality video data of language in (inter)action. Traditionally, this data type is captured using either a mobile eye-tracker or a fixed camera. In a following step, the data need to be annotated in terms of relevant parameters related to speech (transcriptions including verbal and para-verbal information such as hesitation markers, pauses, intonational contours, etc.), bodily action (including hand gestures, body posture, etc.) and, if available, gaze date w.r.t. the aforementioned parameters.

For the majority of studies and available resources (corpora & databases), the annotation work was done manually, based on existing multimodal coding schemes (see [9, 21, 79] for overviews of such schemes). This is a labour-intensive process, requiring multiple annotators and thus substantial resources. As already mentioned as a general guideline, it is assumed that the annotation of a recording has a time-ratio of at least 10:1, thus one minute of video material takes up a minimum of 10 minutes of manual annotation, depending on the level of detail required. When a detailed transcription is needed, in combination with the annotation of several multimodal layers, this ratio can easily amount to 50:1, which makes the manual compilation of large-scale annotated data sets practically unfeasible. Due to this issue of labour intensiveness, some large scale-databases designed for multimodal analysis have not been annotated yet, as is the case e.g. for the Distributed Little Red Hen Lab[1]. For projects such as these, a (semi-)automatic approach to at least some steps in the annotation process is essential. Only by introducing such an approach, the wealth of available data can be made accessible for multimodal analysis.

Building on our hand detection approach as proposed in the previous chapter,

[1] `http://www.redhenlab.org/`

we present an automatic tool for the segmentation and annotation of specific gestural features during language production. The purpose of this tool is to provide a reliable basic annotation that can be easily enriched by further manual analysis. The proposed gesture analysis algorithm produces a segmentation of gesture and non-gesture sequences, spatial information of each gesture based on McNeill's model [93], and an indication of the directionality of each gesture. For each of these dimensions, an XML file is generated, which makes the output compatible with multimodal annotation tools such as ELAN or ANVIL, and integretable with existing transcriptions and annotations.

To maximise the applicability of our approach, the system must deal with the following challenges: (i) The system should be maximally unobtrusive, i.e. it should work on existing video data without information collected with markers or other sensors. (ii) The system should be able to work on videos captured by either a fixed camera or a wearable camera such as a mobile eye-tracker worn by an interlocutor. Furthermore, the participant must have the ability to move freely, thus no markers can be attached to the participant. (iii) The accuracy of the resulting annotated gesture sequence should be very high, requiring virtually no manual correction afterwards.

To summarise, our goal is to develop a highly accurate, automatic gesture analysis tool that is applicable on recordings in which a minimum of restrictions are imposed on the participants. Using such an approach will benefit the analysis of this type of data. In general, the analysis of gesture in human interaction is based on either experimental data or on relatively small-scale corpora. The experimental data have the advantage that the researcher can control some of the variables, making it easier to elicit rich data (and thus to manually process the collected data). The drawback is (i) that researchers need to design new experiments for each new research question, and (ii) that it is notoriously difficult to elicit naturally occurring interaction in a controlled lab setting. The second method, using video corpus data of spontaneous interactions, obviously reduces the latter risk, but comes with a cost as well. Naturally occurring data may confront the researcher with relative data scarcity (low density of the phenomenon under scrutiny in the data), thus requiring the collection of a large corpus to be able to make well-founded claims. Given the above-mentioned challenges of labour-intensiveness, multimodal corpus studies based on *big data* are scarce, especially in comparison to the strong quantitative corpus movement that can be observed for written language. A semi-automatic annotation procedure like the one presented in this chapter can pave the way for a more quantitative approach.

For validation of our gesture analysis approach, we deliberately chose for pre-annotated recordings rather than annotating the recordings ourselves, since we did not have any background in this type of complex annotation. Furthermore,

(a) Frame of NeuroPeirce corpus

(b) Frame of the SaGA corpus

Figure 6.1: (a)Example frame of NeuroPeirce corpus [20]. (b) Example frame of SaGA corpus [88].

by comparing our automatic gesture analysis approach against professional annotations, we get a better insight into the accuracy and efficiency of our approach. To the best of our knowledge, there are no mobile eye-tracking recordings in which the gestures of an interlocutor are annotated. In fact, it was even challenging to find recordings in which both spatial and temporal parameters of gestures were annotated. We present our results on two subsets of existing corpora: the NeuroPeirce corpus [20] and The Bielefeld Speech and Gesture Alignment Corpus (SaGA) [88]. In each recording, a single person is visible during a natural conversation, as shown in figure 6.1. Each recording was made using a camera at a fixed position. The annotations of both recordings include gesture sequences, usage of gesture space, speech, etc.

On top of this validation, in chapter 7, we will analyse a large-scale mobile eye-tracking recording of a human-human interaction experiment using our gesture analysis and compare the annotations against manual coding.

6.2 Related work

This section presents an overview of existing algorithms and approaches relevant to the field of gesture and interaction analysis. Since the concept of automated gesture analysis is a general term, we narrow down the focus of this chapter as the detection of gestures in standard 2D colour camera images. It is important to highlight the difference between gesture *detection* and gesture *recognition*. Gesture detection essentially involves segmenting gesture sequences from non-

gesture sequences. Gesture recognition, on the other hand, requires the re-identification of specific gestures. In this chapter, the focus is primarily on gesture detection as a first, but essential, step in any gesture annotation process. Moreover, there is also a difference between fine-grained finger movements with specific semantics, as is the case in sign languages, and the larger movements of the entire hand as in pointing or waving. Here, we focus on the larger movements rather than the detection of individual finger poses, again as a first, but necessary, step towards even more fine-grained systems. In what follows, we present an overview of existing techniques regarding various aspects of gesture analysis. The following subsections are organised as follows: first we describe some state-of-the art approaches for fine-grained finger movements, then we give a short overview of approaches that combine several modes and finally, we give an overview of techniques that work on traditional 2D images.

Fine-grained finger movement

Rautaray and Agrawal [108] present a thorough overview of existing approaches to the analysis of fine-grained finger gestures. In this survey several publications on vision-based hand gesture recognition for human-computer interaction were identified and discussed. In another recent work, Badi [12] proposes a novel method for the recognition of six specific hand poses in the context of human-computer interaction (HCI): open, close, cut, paste, maximise and minimise. Input images are standard RGB images, but severely conditioned: containing only a single hand on a black background. Two features are extracted from these images: hand contours and complex hand moments. In a final step, these features are used in an Artificial Neural Network (ANN) classifier to identify the different hand poses. Another approach to the recognition of specific finger poses can be found in [75]. Here, two types of depth cameras, viz. Time-of-Flight (ToF) and Kinect, are used to recognise hand gestures. Next to the identification of dynamic gestures such as “to feel” or “to ache”, they also developed a technique for the recognition of the specific signs for the Polish finger alphabet. In this system, two classification approaches are compared: a Hidden Markov model (HMM) classifier and a nearest neighbours technique with dynamic time warping (DTW), allowing a non-linear mapping of one pose to another by minimising the distance between them. A similar approach is found in [84], where a highly precise method for the recognition of static hand gestures is proposed using data from a consumer depth camera. In addition to the approach of Kapuscinski et al. [75], a multi-layered random forest (MLRF) classifier is used to identify different signs such as the 24 letters of American Sign Language (ASL).

It is important to note that in the techniques described above, the input

data contains information of a single hand, either in a standard image or in depth information. In our approach on the other hand, we are interested in larger movements of the entire upper body. Here, the input data traditionally contains footage of an entire person in a much more natural setting and thus, an additional challenge consists in the segmentation of relevant body parts from the background. In the next subsection, several existing approaches to this type of gesture analysis are discussed.

Hand detection by combination of multiple modes

A recent challenge that addresses gesture recognition for larger movements is the *Chalearn looking at people challenge* [48]. In this challenge, several modes can be utilised to automatically recognise a vocabulary of 20 Italian cultural/anthropological signs in image sequences. These modes include RGB images, depth images, skeleton representation and binary masks. As expected the top-competitors [28, 99, 102] of this challenge combine several modes to achieve top accuracy. Another approach in which RGB, depth and data from a motion capturing system (Xsens) are combined to locate the position of both hands was developed by Yin and Davis [140]. They used an off-the-shelf skin segmentation to mask the Kinect depth data. Once the position of the hands is found in each frame, a HMM is trained for each phase for each gesture. This means that a separate model was trained for pre-stroke, nucleus and post-stroke. In the end, a Viterbi decoding was used to optimally segment the gesture sequences. The above-mentioned approaches are capable of detecting gesture sequences and they can identify specific gestures. However, since not all existing recordings are captured using multiple and/or specific cameras, our goal is to develop a system that is able to detect gestures in normal RGB images in natural settings. In the next subsection we discuss existing approaches that only rely on RGB data.

Hand and gesture detection in RGB images

In case a recording was made without a depth sensor nor motion tracking system, the level of complexity of gesture detection increases significantly. Due to the lack of this additional depth and skeleton information, one needs to detect the relevant body parts in advance and thereafter one could start detecting the gesture sequences. In general, gesture analysis in standard RGB images consists of two main phases: first the retrieval of the hand positions, and secondly, the segmentation of the recording in gesture and non-gesture sequences using extracted information from the hands. For an overview of techniques that detect hands in images, we refer the reader to section 5.2.

Once the location of both hands is found in each image of a recording, a gesture detection algorithm can be applied. The purpose of such an algorithm is to segment a recording in gesture and non-gesture sequences. In [56] an approach for the automatic detection of gesture strokes was presented. Next to the skin segmentation, they also apply a corner tracking algorithm to the segmented image. Their approach is developed to cluster three sets of corners: one cluster for each hand and one for the head, assuming there is only one person present in the video. Finally, values extracted from the clusters of corners are fed into a machine learning algorithm that is trained to predict whether or not a given frame is inside a stroke. Unfortunately, they achieve an average accuracy (F_1-measure) of only 38.71%, which makes their approach insufficiently accurate for practical use. Another gesture spotting system is presented by Peng [105]. Here, a simple yet effective approach to divide a video in short clips of gestures and non-gestures was proposed. They assume that the hands of an actor are almost in the same position when he or she is not performing a gesture. Using this assumption, one could determine a static hand position for each hand. By performing a frame-based calculation of the distance between each hand and the static hand positions, one could easily distinguish the gestures from the non-gestures. Furthermore, they apply a gesture recognition step to classify various gestures. However, although they achieve high accuracy in terms of recognising different gestures, they do not provide any measurements of the detection of gesture sequences. Since they define a static position for each hand, their approach can only be applied in a context with a fixed camera and an immobile subject. Schreer and Masneri [116] presented an automatic video analysis for the annotation of human body motion in humanities research, which is highly similar to our goal. The first step in their approach consists of a skin colour segmentation that is done manually using a set of sliders. After this skin segmentation, their software tracks the hands based on their motion. This motion information is used to segment the video in gesture and non-gesture parts. On top of the detection of gesture sequences, their tool also provides information regarding the type of movement in terms of: Phasic, Repetitive and Irregular. Next to the gesture sequence detection, they also provide automatic information on the position of the hands related to the body as defined in the McNeill gesture space [93]. Despite their efforts, the accuracy of the gesture sequence detection on their datasets is limited to 75.3%. In our opinion, another drawback of this approach is the skin segmentation using a set of sliders, making the accuracy of their approach unpredictable.

It is clear that automatic gesture detection, although it has been studied for several years, remains a hot topic in several research fields. Many approaches rely on multiple methods to detect the gesture sequence. In the above-mentioned papers, some novel methods for the detection of gestures are presented. Unfortunately, none of them meets our imposed requirements in

terms of applicability and accuracy i.e. maximally unobtrusive, ability to handle moving camera viewpoint, and achieve top accuracy on challenging recordings. Nevertheless, we used some of the previously described concepts in our implementation to develop the gesture detection algorithm. For example: taking into account the distance between a hand and its static position as proposed in [105] and expressing the usage of the gesture space according to the McNeill definition as proposed in [116].

6.3 Approach

As previously mentioned, the first step in a gesture analysis algorithm consists of retrieving the positions of the hands in each image of a recording. For this purpose, we chose our own segmentation-based hand detection approach, as presented in the previous chapter. This approach has the advantage that it is highly accurate, since we provide the opportunity to manually steer the detections when necessary.

Once the positions of the hands in an entire recording are retrieved, the automatic gesture analysis can be initiated. As discussed in section 6.2, there are several approaches for this type of analysis. Despite the large variety in approaches, we propose the development of a gesture analysis tool in which we impose a minimum of restrictions to the participants in order to enlarge the applicability. For example, our gesture analysis tool should be able to handle participants in a sitting as well as standing position. On top of that, we allow a participant to walk during the recording and we even allow a moving camera position, which is particularly useful for recordings made using mobile eye-trackers. Finally it is important to notice that our gesture analysis tool relies only on the semi-automatic retrieved hand detections as described in the previous chapter (i.e. detections of the hands and the position of the human upper body and face). No additional information such as depth information or motion sensors is required.

Our gesture analysis tool consists of several blocks, as shown in figure 6.2. Each individual block is described in the following subsections, starting with the calculation of the rest position in subsection 6.3.1. Once the rest positions are known, we segment a recording in gesture segments and non-gesture segments based on the displacement between each hand and its respective rest position as described in subsection 6.3.2. Subsection 6.3.3 discusses our approach to automatically provide information concerning the gesture space. A final analysis is applied on the directionality of the gestures as given in subsection 6.3.4.

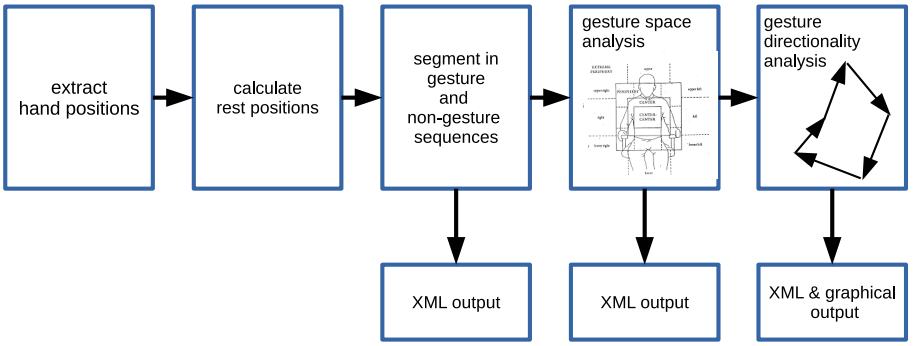


Figure 6.2: Workflow of our automatic gesture analysis tool. This figure also reveals which type of analysis results in XML compatible output.

6.3.1 Rest position

Starting from the hand positions, there are two main approaches to segment a recording into gesture and non-gesture sequences. The first method monitors the velocity of the hands with the assumption that a hand does not move when one is not gesturing, as is proposed in [116]. A second approach measures the distance between a rest position (i.e. the position of the hands when one is not gesturing) and the hands as proposed by Peng [105]. Despite the fact that both approaches are widely applied in the literature, we argue that the second one is more generic. Indeed, during our experiments we noticed that, for particular large gestures, the velocity of the hands stalls within the gesture. An obvious example is the hold phase of a pointing gesture. Here, the first-mentioned approach would not recognise the hold phase as (part of a) gesture, since there is little or no velocity of the hand.

Essential for the chosen approach is the determination of the rest positions. We can define this position as: *the position were the hand is located most frequently during an entire recording.* An obvious approach is simply plotting the positions of both left and right hand into a map and calculating a local maximum for each hand. In our application on the other hand, we allow moving participants as well as a moving camera viewpoint, which means we need to transform the coordinates of the hands relative to the position of the participant. Since we use the detections that are retrieved using our semi-automatic approach, we have access to the coordinates of the human upper body detection in each frame to accomplish this task. The transformation of the hand coordinates is given in equation 6.1, where x_H^{rel} represents the relative x-coordinate of the hand, x_H stands for the original x-coordinate of the hand, x_U the centre of the human

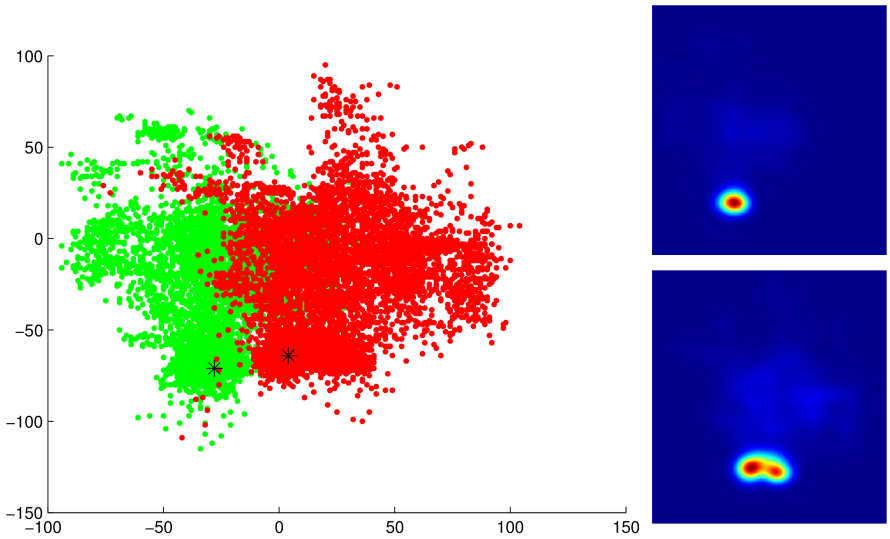


Figure 6.3: Normalised hand positions of an entire recording. Green dots represent right hands, red dots represent left hands. Two asterisks indicate the respective rest positions.

upper body detection and w_U width of the upper body detection. The same methodology is used for the y-coordinates.

$$x_H^{rel} = \frac{x_H - x_U}{w_U}$$
$$y_H^{rel} = \frac{y_H - y_U}{w_U} \tag{6.1}$$

By applying this transformation to each frame of a recording we obtain a map as shown in the left part of figure 6.3. The two asterisks represent the local maxima for both left and right hand. These are indeed the rest positions for each hand for that particular recording. They are obtained by extracting the local maxima for each hand in a Gaussian smoothed map as shown in the right part of figure 6.3. The upper smoothed map belongs to the right hand coordinates, the bottom smoothed map belongs to the left hand coordinates. Here the colour represents the density of the hand coordinates: red means dense coordinates, whereas blue means sparse coordinates. Once the rest positions are known, we are able to segment an entire recording in terms of gesture sequences and non-gesture sequences as described in the next section.

6.3.2 Gesture segmentation

The segmentation of gesture and non-gesture sequences is done by calculating the displacement between each hand and its respective rest position. Once the displacement of at least one hand is beyond a set threshold we assume that a gesture sequence was initiated. When subsequently the displacement drops below this threshold, the gesture sequence ends since the corresponding hand is back in the vicinity of the rest position. Using this methodology, we are able to segment an entire recording in gesture and non-gesture segments.

A decisive aspect in this approach is the calculation of the optimal threshold value. This value indicates how much deviation from the rest position is allowed before our software initiates a gesture sequence. A straightforward approach is to define a fixed value that is used for both hands. However, our experiments revealed that it is challenging to find a unique value that results in accurate segmentation of multiple recordings each with its own characteristics. To overcome this problem, we proposed a set of solutions: a) separate thresholds for both left and right hand, b) separate threshold values in both x and y direction, and c) obtain threshold values from the data itself by extracting a sigma from the Gaussian smoothed hand maps, as shown in the right part of figure 6.3. An illustration of this technique for the right hand in a recording is given in figure 6.4. Here we plot, for the first 1000 frames of a recording, the Euclidean distance between the right rest position and the assumed right hand. The red line in the top part represents the applied threshold. Indeed, for simplification we used a single threshold in this example rather than a unique threshold for x and y direction. In the bottom part, we illustrate the gesture segmentation based on this displacement. In our framework, an additional temporal smoothing is applied to reduce jitter and to cluster neighbouring gesture segments.

$$\begin{aligned} GS = \{(D_{LX} > \alpha\sigma_{LX}) \vee (D_{LY} > \alpha\sigma_{LY})\} \vee \\ \{(D_{RX} > \alpha\sigma_{RX}) \vee (D_{RY} > \alpha\sigma_{RY})\} \end{aligned} \tag{6.2}$$

In equation 6.2 the condition to initiate a gesture sequence GS is given. D_{LX} represents the displacement of the left hand in x-direction, σ_{LX} is the sigma value for the left hand in x-direction that was obtained from the smoothed map and finally α is a tuning parameter that can be used to fine-tune our system. If at least one of the four displacements exceeds its respective threshold, a gesture sequence is initiated. The accuracy of this segmentation is thoroughly discussed in section 6.4.

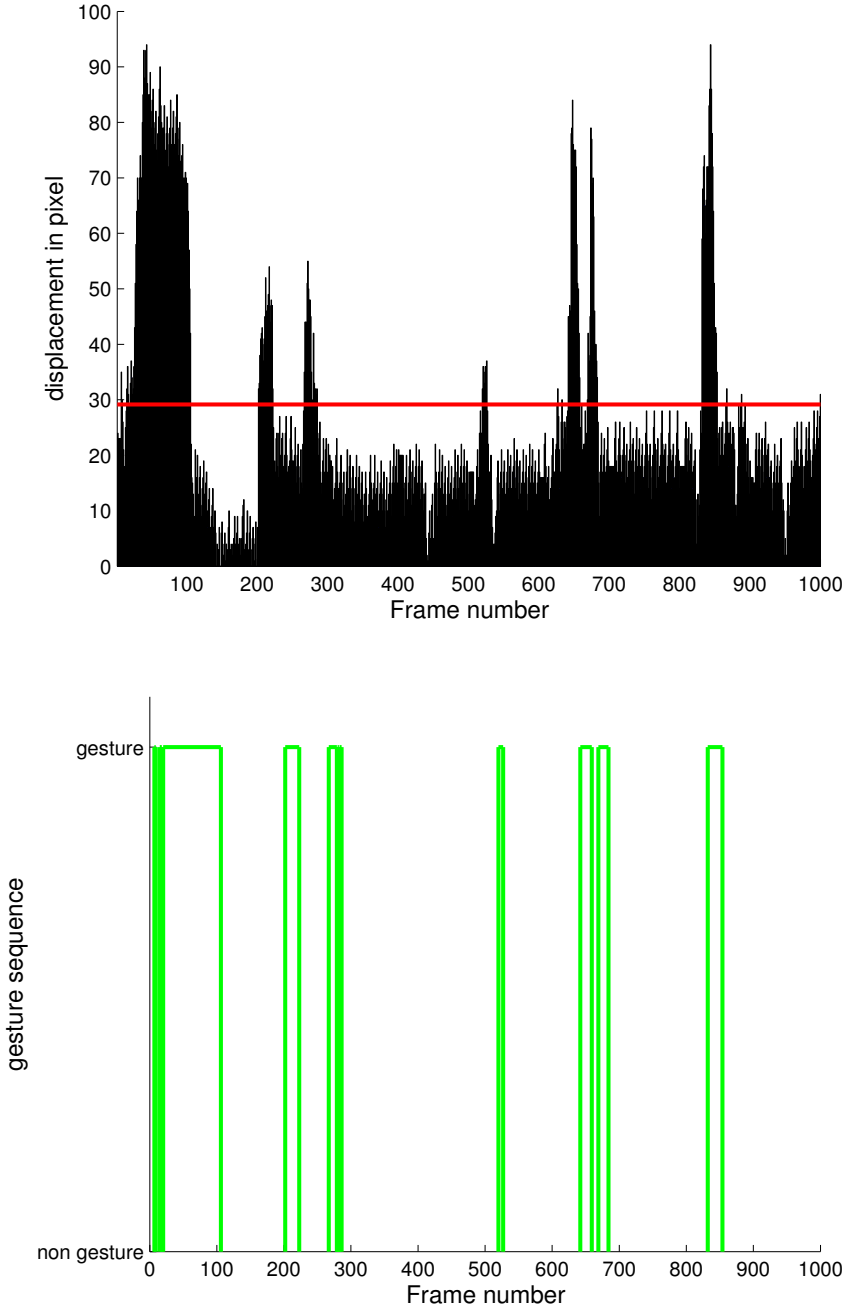


Figure 6.4: Top part: displacement of the right hand w.r.t. the rest position. Red line indicates the applied threshold. Bottom part: gesture segmentation that is generated using this displacement.

As a final remark, it is important to mention that this analysis is automatically written into an XML file, containing the gesture segmentation annotations. Next to this basic segmentation, our gesture analysis tool also generates information regarding the usage of the gesture space as described in the next subsection. This file type is compatible with existing annotation tools such as ELAN or ANVIL.

6.3.3 Usage of gesture space

Researchers in gesture studies are interested in the spatial distribution of gestures, i.e. where in the gesture space a gesture occurs. A commonly used methodology for this purpose is the gesture space as defined by McNeill [93] and illustrated in figure 6.5. He proposed to divide the space into sectors using a system of concentric squares. The sector directly in front of the chest is the center-center sector. Surrounding this, the center sector is defined. Then the periphery, which is subdivided into upper, lower, right and left. Finally, the extreme periphery is defined, which is divided into even more sub-sectors. Manually annotating the gesture space is extremely labour-intensive, since ideally, one has to assign a specific gesture sector to each individual frame of a gesture sequence. In order to reduce this workload, we noticed that the manual analysis of the gesture space is often reduced to the allocation of a single sector for each entire gesture. For example: the annotation of the sector where the majority of the gesture occurs or the annotation of the sector where the gesture is the largest. It is clear that such an annotation reveals only a fraction of the spatial information.

In order to overcome this problem, we can use additional data, which is obtained by using our semi-automatic hand detection approach, to automatically annotate the gesture space. As mentioned in section 5.4.2, next to the hands, both face and upper body are detected. We defined a mathematical relationship between the face detection, upper body detection and the individual gesture sectors as defined by McNeill. This allows us to automatically define the gesture sectors on each individual image as shown in figure 6.6. Here, we distinguish the four larger sectors: center-center, center, periphery and extreme periphery as well as the subdivisions for both periphery and extreme periphery. Using these automatically generated sectors, we are able to easily determine in which sector each hand is located at each moment. Therefore, our system automatically compares the hand coordinates with the coordinates of each sector. Similar to the previously mentioned approach, this analysis is also stored in an XML file. For each hand a unique tier is added in which, for each gesture sequence, the usage of the sectors is annotated. Compared to manual analysis, our approach always provides a frame-based analysis of the gesture space, which is in case of

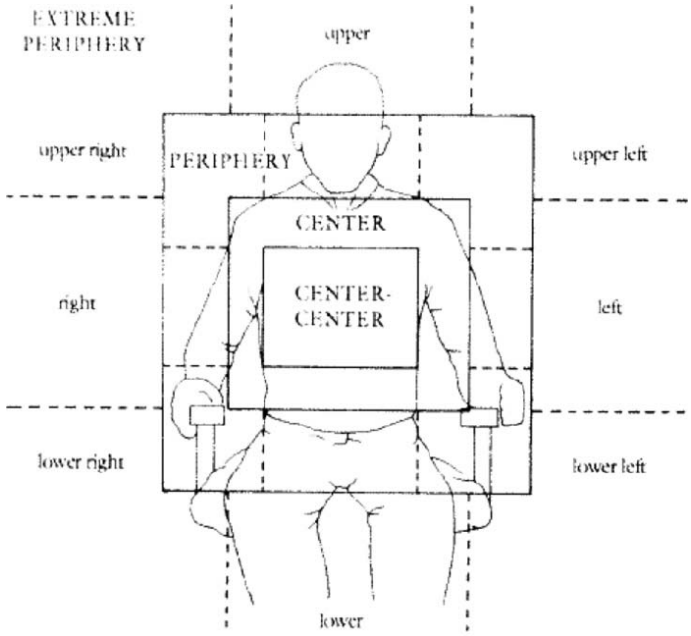


Figure 6.5: Gesture space as defined in [93]. We can distinguish 4 larger sectors as represented by capital letters as well as the respective sub-sectors.

manual analysis practically infeasible. Since our automatic analysis generates gesture sectors based on both face and upper body detection, we are able to ensure a consistent and non-subjective definition of the sectors across several recordings. In manual labelling on the other hand, significant differences exist in the exact definition of the sectors between several annotators. Our automatic system excludes these unwanted side effects.

6.3.4 Gesture directionality

A final type of analysis to be included in our system is the directionality of gestures. Using the above-mentioned approaches, we are able to automatically segment a recording in gesture and non-gesture sequences. Furthermore, we can automatically provide information regarding the gesture space for each individual frame. Another vital aspect of gesture analysis is the directionality of gestures. Researchers are interested in the direction and movement of each gesture, resulting in a specific trajectory of each hand (see e.g. the gesture

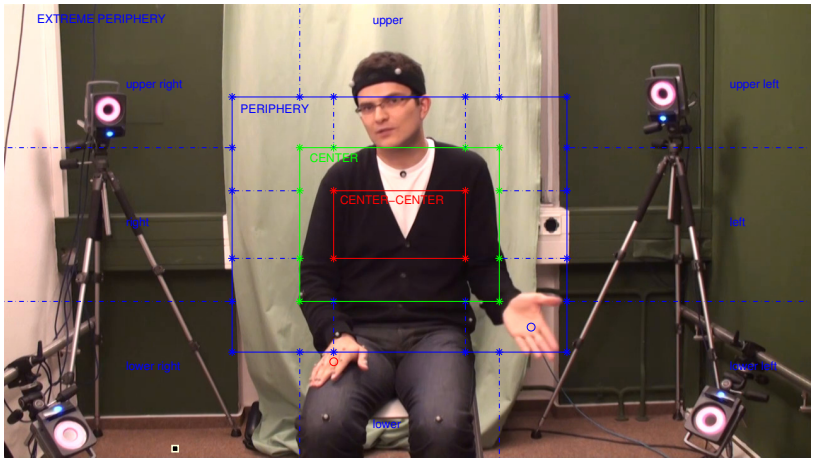


Figure 6.6: Automatically generated gesture space based on the upper body and face detections. Here the left hand is located in the *periphery*, while the right hand is located in the *extreme periphery*. Image obtained from the NeuroPeirce corpus [20].

annotation scheme presented by Bressem [21]). Although this is of particular importance in several aspects of gesture analysis, we often notice that manual annotation is restricted to a partial analysis. For example, the directionality of an entire leftward pointing gesture is often annotated as *left*, since this is the major direction of movement. Again, this partial analysis arises from the tremendous amount of work that manual, frame-based, analysis requires. To further support the annotation of this type of recordings, we propose an automatic alternative. Here, we calculate the direction of movement for each frame by comparing the hand positions of the current frame and the positions in the previous frame. Thereafter we apply a temporal smoothing by using a 1D convolution filter to reduce jitter on the annotations. Our approach automatically annotates four directions: left, right, upwards and downwards for each hand in each frame of a recording. And again, the results of this analysis can be exported to an annotation file for further analysis.

As mentioned before, the focus of this section was on the accurate detection of gesture sequences rather than the recognition of specific gestures. Next to this gesture detection, we also extract relevant features from each gesture sequence such as usage of the gesture space and directionality of the gestures. Since our goal was to develop a tool for simplifying the work of manual annotators, our system needs to achieve high accuracy. Therefore, we performed a profound

validation of the above-mentioned approaches. In the next section, we present the results of this validation in terms of accuracy, usefulness and cost-effectiveness compared to manual analysis.

6.4 Results

To validate our approach, we searched for existing pre-annotated corpora as a basis for comparison between manual and semi-automatic annotation. We found two independent institutes willing to provide their corpora as well as their annotations. The first corpus, the NeuroPeirce corpus [20], was created by the research group of Professor Irene Mittelberg (University of Aachen) in the context of a larger research project. Within this project several recordings were made of participants during natural gesturing i.e. non-elicited speech and gesture production. For our validation, we got access to one recording of approximately 7 minutes (10500 frames) as well as the corresponding annotations in ELAN. These annotations include the above-mentioned parameters of gesture phases, position in gesture space, etc. The second corpus was created at the university of Bielefeld and is known as *The Bielefeld Speech and Gesture Alignment Corpus* (SaGA) [88]. Here direction-giving dialogues were recorded using multiple cameras. For our validation we used a recording lasting approximately 8.5 minutes (in total 13000 frames). Again, this recording was annotated in ELAN and includes annotations of the gesture phases. Example frames of both corpora can be found in figure 6.1 in section 6.1.

We processed each recording using the above-mentioned gesture analysis approaches, but first, we used our semi-automatic segmentation-based hand detection tool of the previous chapter to retrieve the hand, face and upper body locations in each frame. Some measurements regarding this initial analysis can be found in table 6.1. Here, we notice that the amount of manual interventions required by the system (based on the predefined threshold) is negligible. In less than 3% of the frames manual annotation was required, i.e. a reduction of manual work with a factor of 37 as compared to fully manually annotating each frame. The total processing time includes the face and upper body detection, the generation of the skin segments, filtering the candidates as well as the manual interventions. Indeed, the total processing time required by our semi-automatic approach is similar to the 10:1 manual annotation time-ratio. However, it is important to notice that when we assume that one click of manual intervention takes about 1 second, respectively only 8 and 10 minutes of manual input was required for the analysis of both recordings. This means that, for the amount of manual work, we almost reach a 1:1 ratio. The remainder of the processing time is spent by the computer on automatic analysis and is therefore

Table 6.1: Measurements of the semi-automatic hand annotation of both recordings.

	NeuroPeirce	**SaGA**
duration of recording	7 min	8,5 min
#frames in recording	10500	13000
#manual annotated frames	253	358
pct manual annotated frames	2,41%	2,75%
time automatic candidate generation	40 min	34 min
time automatic filtering candidates	29 min	28 min
amount of manual intervention	8 min	10 min
total processing time	77 min	72 min

not labour-intensive. Furthermore, this (semi-)automatic analysis consists of several layers, since both hands, faces and upper bodies are detected. This implies that the manual annotation of these recordings would take more time than the basic 10:1 time-ratio.

Starting from the hand, face and upper body detections in each frame, we can apply our gesture analysis tool to both recordings. In the remainder of this section, we compare our automatic analysis to the manual gesture annotations. First, in section 6.4.1 we evaluate the accuracy of the semi-automatic hand annotation tool, after which we present the results of an accuracy measurement of the gesture phase segmentation in section 6.4.2. In a third 6.4.3 and fourth 6.4.4 section, we review the gesture space annotation and directionality.

6.4.1 Accuracy of the hand annotations

Since the accuracy of our gesture detection algorithm relies on the accuracy of the semi-automatic hand annotations, it is important to validate their accuracy. For this, we manually labelled the hand positions in the first 1000 frames of the NeuroPeirce recording. Then, we compared our semi-automatic hand annotations to this ground-truth in terms of accuracy. Similarly to the methodology used in the previous chapter, a hand annotation is considered valid if the distance between the detection and the manual annotation was below half face width. This comparison revealed that in 97% of the hands, the position was obtained correctly using our semi-automatic approach. Furthermore, for the entire set of 2000 annotated hand positions, the average distance error was only 10 pixels, which indicates that our hand-detection approach is indeed highly accurate and can be used as a basis for our gesture analysis tool.

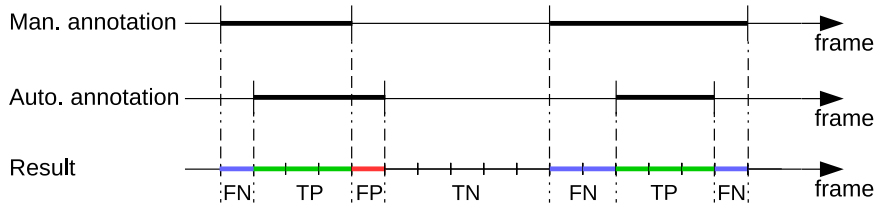


Figure 6.7: Validation methodology that is used for the gesture phase segmentation.

6.4.2 Accuracy of gesture phase segmentation

To validate our gesture phase segmentation, we propose a frame-based comparison between our automatically generated gesture sequences and the manual annotations of both recordings. Based on the manual gesture phase annotations in ELAN, we assigned a label to each frame: **1** if the frame was part of an annotated gesture phase, **0** otherwise. The same frame-based information was extracted from our automatic gesture phase segmentation. Finally a validation scheme as illustrated in figure 6.7 was used. For each frame we compare manual and automatic annotations, resulting in one out of four labels per frame: True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Using these labels, we can determine the accuracy of our approach as shown in equation 5.8 in the previous chapter. By combining both precision and recall into the (F_1)-score, we obtain a single value that expresses the accuracy of our system. It is important to notice that, however the validation of the hand annotations was done on the smaller subset of 1000 frames, the remainder of the accuracy experiments are performed on the entire NeuroPeirce and SaGA recordings.

In equation 6.2, we already introduced the tuning parameter α. This parameter is used to find the optimal fraction of the σ thresholds. For validation, we varied α in the range from 0 up to 3 in steps of 0.1 in order to find the optimal setting. The results of these experiments are shown in figure 6.8, in which we plot precision versus recall. The most optimal point on such a graph is the upper-right corner (both precision and recall equal to 1). It is clear that the curves of both recordings approach this point. Both curves reach their best accuracy at the same α: 0.9. The corresponding accuracy measurements for this α are given in table 6.2. Here, we notice that our approach achieves very high accuracy on both recordings.

Next to the validation of our own approach, we also compared our accuracy against another gesture analysis algorithm. We opted to use the AUVIS gesture

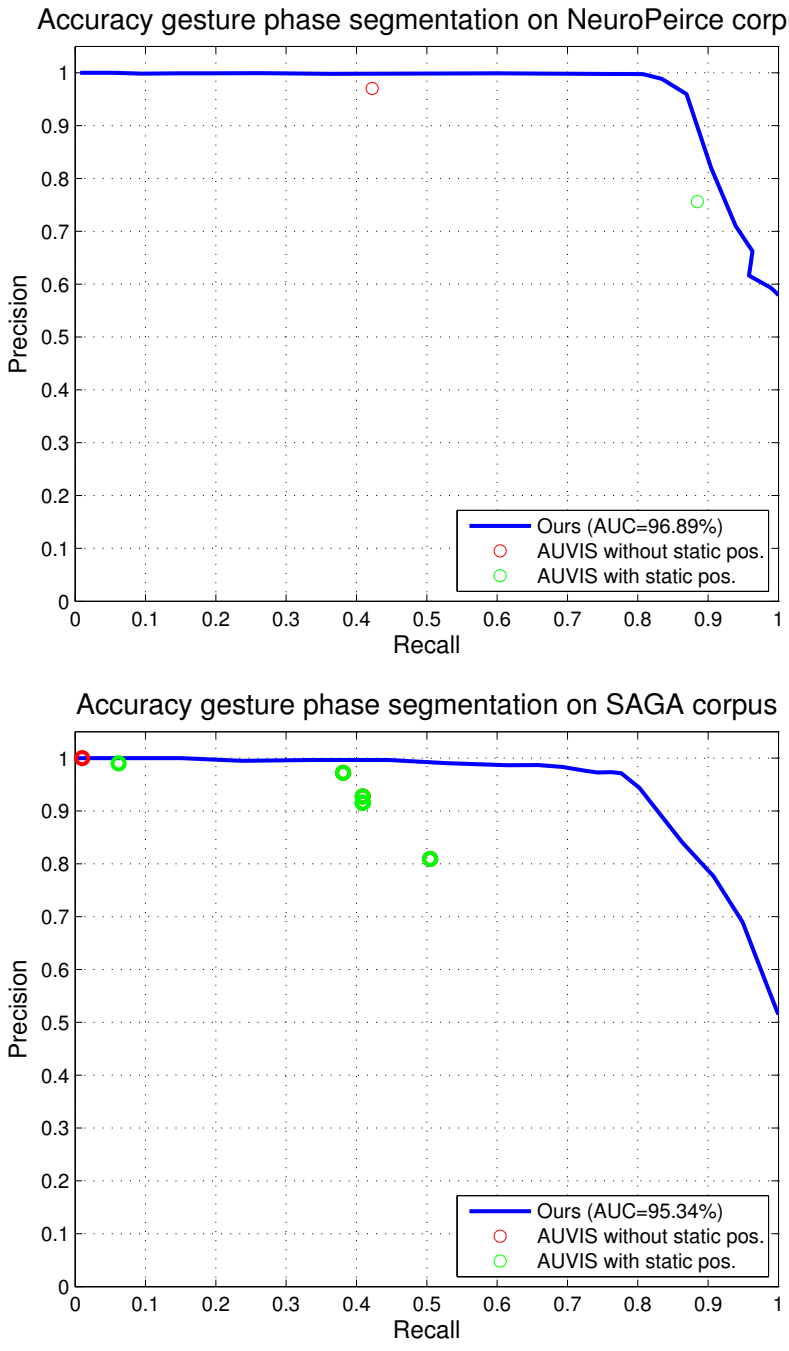


Figure 6.8: Precision-recall curves of our approach for both recordings. Coloured circles represent the accuracy of the AUVIS [116] method.

Table 6.2: Accuracy of the optimal working point i.e. $\alpha = 0.9$. Bottom row of the table shows the best accuracy of the AUVIS tool tested on the same corpora.

	NeuroPeirce	**SaGA**
Precision	98,85%	94,38
Recall	83,41%	80,22
F_1-score	**90,48%**	**86,73%**
Best F_1-score AUVIS [116]	82.23%	62.17%

analysis tool as presented by Schreer and Masneri [116], since a) it formulates the same goal and b) it is directly available in the ELAN annotation software. For more information regarding this integration, we refer to [2]. As mentioned in section 6.2, their approach relies on manual tuning of the skin-segmentation parameters. We asked 5 participants to perform this tuning for each recording in order to provide a fair comparison. These participants were both experienced and non-experienced annotators. In this publicly available implementation of their approach, two methods for gesture segmentation are available: taking the distance to the static position into account or not using the static position and only relying on the velocity of the hands. The results of their approach are shown in figure 6.8 using the coloured circles.

These circles reveal that the skin segmentation of the NeuroPeirce recording was relatively easy, since the same accuracy was achieved by each of the five skin segmentation settings. On the other hand, the skin segmentation of the SaGA recording was far more complex as shown by the diverse accuracy results. This was mainly caused by the presence of the wooden chair in each image as shown in figure 6.1, which has more or less skin tone. In table 6.2, the best F_1-score of the AUVIS approach is given for both corpora. Overall, it is clear that our approach outperforms the method of [116]. Furthermore, our approach results in more consistent accuracy over multiple recordings without time-consuming and subjective parameter tuning.

6.4.3 Accuracy of gesture space annotation

The validation of the gesture space is far more complex since the available annotations are inadequate. As mentioned above, the majority of existing gesture space annotations cover only a small portion of the data. Indeed, the annotation of the gesture space in the NeuroPeirce recording was restricted to a single label for each gesture stroke, whereas our system provides a frame-based gesture space annotation for an entire gesture phase (preparation, stroke and

retraction). This imbalance in level of precision made it difficult to perform a meaningful comparison between the manual annotation and our automatic labelling. The SaGA recording did not include annotations of the gesture space at all.

Since our gesture space analysis provides a frame-based annotation, we needed to transform this output for validation. We extracted the frame sequences from each gesture stroke and calculated the most occurring label in each stroke. Then, we compared these extracted labels to the manual annotations, resulting in an accuracy of 64.42%. Again, this comparison is suboptimal since our automatic analysis produces a much more fine-grained annotation compared to the manual annotations.

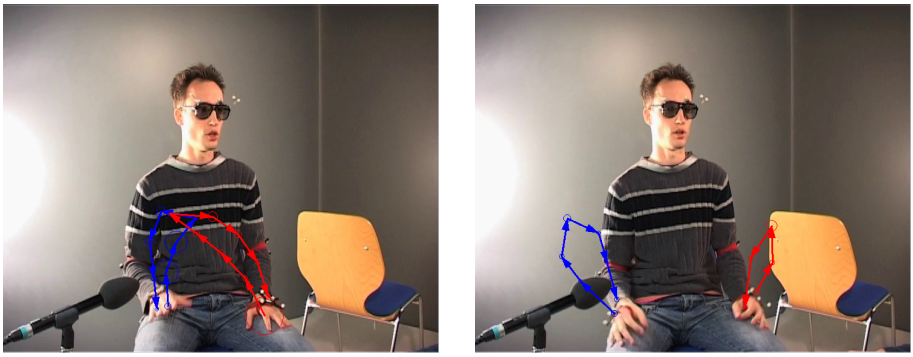
Since there is a mathematical relation between the hand positions and the gesture sectors, one can assume that if the positions of the hands are obtained highly accurately, our gesture space analysis is likewise accurate. As mentioned above, the accuracy of our semi-automatic hand annotation tool is 97%, thus we might suppose our gesture space annotation is as accurate as well.

6.4.4 Output of gesture directionality

In this last subsection, we discuss the results of the analysis of gesture directionality. As mentioned in section 6.3.4, our system defines the direction of each hand in each frame. Comparing our automatic direction labels to the manual annotations was impossible since none of the corpora provided such annotations, probably because it is practically unfeasible to perform this type of annotations manually. Nevertheless, we might assume that our directionality analysis is highly accurate, since it is directly extracted from the semi-automatic hand annotations.

A final advantage of this automatic frame-based analysis is the possibility to represent an entire gesture into a single frame. Examples are given in figure 6.9. Here we show the first frame of each gesture sequence and we plot a circle for the individual hand positions of the entire gesture onto this frame. The displacement between two frames is illustrated using arrows. In case a hand was held still, this is indicated by increasing the radius of the corresponding circle. Such a representation of each gesture can be used in a graphical representation of a recording, where for example the gesture images represent relevant moments on a timeline.

At last, it should be noted that the total processing time of our entire gesture analysis algorithm amounts to only a few minutes to analyse a video recording of 10 minutes, since it exclusively consists of processing the detection file as



(a) (b)

Figure 6.9: Examples of gestures captured into a single image.

obtained using our semi-automatic hand detection approach. This implies that the entire analysis in which hands, faces and upper bodies are detected, as well as a gesture analysis including gesture segmentation, usage of gesture space and gesture directionality, is performed in a time ratio of approximately 10:1. As already mentioned, on average, only 10% of this analysis time involves manual analysis. Next to the analysis of these traditional recordings, which were made by a fixed camera, our approach is obviously also applicable to the analysis of mobile eye-tracking recordings. In that case, an additional analysis is performed, viz. mapping the gaze data on top of each analysis item. Thus, our approach automatically segments the gestures that are recorded using the scene camera of the mobile eye-tracker and automatically detects to which gestures the subject spent visual attention. It is clear that the manual analysis of such a recording is extremely complex and therefore time-consuming.

6.5 Conclusion

In this chapter, we presented an automatic approach to the annotation of gestures that are relevant in research on human-human interaction, as e.g. in the field of psychology, linguistics or behaviour analysis. Our focus lies on minimising the workload that is related to this type of annotation. To provide a useful alternative, it is of vital importance that our approach produces highly accurate annotations. Therefore, our first step is the extraction of relevant body parts including face, human torso and hands in each image of the recording. This is achieved using our own semi-automatic hand detection algorithm. Several

validation experiments revealed that this method is capable of reliably detecting the hands, which makes it a valid basis for gesture analysis. On top of that, we are able to reduce the amount of manual analysis by a factor of 37 as compared to fully manually annotating each frame. The gesture analysis starts by defining the rest position of each hand during an entire recording. Once this location is known, we calculate the distance between each hand and its respective rest position for each frame. Based on this displacement, we are able to segment a recording in gesture and non-gesture segments. On top of that, we automatically analyse the usage of the gesture space according to the McNeill [93] sectors. A final analysis is done on the directionality of the hands. Here, we analyse the trajectory of each hand during gesturing. Each analysis generates a unique tier in an XML compatible file, making our approach integratable with existing annotations. We performed a thorough comparison between our automatic gesture analysis and the annotations of two existing corpora revealing that our approach is highly accurate.

Similarly to the previous chapters, we use the proposed gesture segmentation tool in chapter 7 for the analysis of a challenging of real-life mobile eye-tracking recording.

Chapter 7

Large scale experiments

The final goal of this dissertation is the development of a framework for the efficient and accurate analysis of mobile eye-tracking recordings that are made in a variety of application domains. In the previous chapters, several approaches were proposed and each of them was validated using a frame-based validation scheme. In this chapter, we compare our automatic analysis approach against manual analysis in terms of accuracy and efficiency. For this purpose, we automatically analyse several challenging real-life, mobile eye-tracking recordings. Comparing our automatically generated annotations against manual labellings will reveal the usefulness and applicability of our approach. Besides this validation, we present various methods for representing and interpreting the results of our automatic analysis.

The remainder of this chapter is organised as follows: in section 7.1, we introduce the traditional methodology that is used for the manual analysis of mobile eye-tracking recordings. In section 7.2, we discuss several methods that are used for expressing the level of agreement between different annotations. In section 7.3 multiple recordings of a customer journey experiment are analysed, while in section 7.4 a recording of a human-human interaction experiment is analysed. Section 7.5 describes the analysis of a recording in which a subject attends a presentation. Finally, in section 7.6, we give an overview of the visualisation methods that we have developed for interpreting and accessing results.

7.1 Introduction

Throughout the previous chapters, we developed various methods for the automatic or semi-automatic analysis of mobile eye-tracking recordings. We start this chapter by giving the reader a short resume of the developed approaches. In chapter 3, a method was developed that automatically detects objects that are in the visual focus of attention of a subject. Our framework was further expanded in chapter 4 by developing a method for detecting how often and for how long the subject looks at another person. Furthermore, we proposed a method that specifies at which person in particular the subject was looking. In chapter 5, two approaches were proposed to detect human hands in images that were captured by the scene camera of a mobile eye-tracker. By mapping the gaze data on top of these detections, we can automatically determine how often the subject looked at the hands of another person. Such an approach is in particular useful in a gesture analysis approach as proposed in chapter 6. Here, a method was developed for segmenting a recording in gesture and non-gesture sequences. Furthermore, for each gesture sequence, additional features were extracted.

In each of the previous chapters, a validation was performed to prove the accuracy of the proposed methods. In most cases, this validation was done on images that were captured using an actual mobile eye-tracker. Each approach was validated using a frame-based validation scheme. This means that we expressed the accuracy in terms of how many instances were classified correctly versus the amount of incorrectly classified instances in a frame-based manner. However, although such a validation methodology gives a clear view of the accuracy of each method, it provides no information on the applicability of our approach in the analysis of real-life mobile eye-tracking experiments. In this chapter, we aim to elucidate this matter.

For the actual analysis of a mobile eye-tracking experiment, we map the gaze data on top of each detected item or body part to find out how often and how long the subject paid visual attention to that particular item. It is important to note that the gaze data may consist of either the actual fixations or frame-based raw gaze locations, since our framework is capable of handling both data-types. Furthermore, we use our own data format for representing this gaze information. Therefore, our approach is applicable to any eye-tracking recording, irrespective of the type of mobile eye-tracker that is used. Currently, we developed conversion wrappers for both Arrington, Tobii and Pupil-Pro eye-trackers, however developing wrappers for other eye-tracking brands is unproblematic.

A common method for the manual analysis of mobile eye-tracking recordings or

video-recordings in general, is using annotation tools such as ELAN or ANVIL. An example of such an annotation tool is given in figure 7.1. Traditionally, such a tool is used to annotate or transcribe a recording in terms of relevant characteristics. For example, speech, gaze, gesture segmentation, gesture direction, etc. are annotated. As shown in figure 7.1, for each annotation type, a unique line is created. These lines are further referred to as *tiers*. The upper tier in this figure represents the gaze information. Traditionally, one manually creates segments in which the subject was looking at a specific object or item. In figure 7.1, the gaze cursor (red dot) is currently positioned at the presentation screen. Indeed, the corresponding segment, as indicated by the time cursor (i.e. the red vertical line) is labelled *S5* i.e. the fifth slide of this presentation. The *GesturePhase* tier, contains the gesture segments. Thus, in this particular example, the speaker performs his 68th gesture in this recording. Of course, such an annotation file may contain much more information, but this example gives a clear view of the content and creation of such a file. It is clear that manually creating and labelling each individual segment is extremely time-consuming and therefore error-prone.

As already mentioned in the previous chapters, the output of our analysis framework consists of an XML-based file that is readable by annotation tools such as ELAN. During the automatic or semi-automatic analysis of an eye-tracking recording, our framework automatically generates the relevant tiers, and automatically creates the segments and associated annotation values. The ability to export the annotations to existing software tools enlarges the applicability of our approach, since many analytical tasks consist of additional tiers amongst gaze or gesture information. For example, in the field of linguistics, the speech is often transcribed manually in ELAN (or other tools) and then merged with the gaze and gesture annotations.

In the next section, we describe how we measure the level of agreement between the annotation labels of our automatically generated segments and manual labellings.

7.2 Statistical analysis

Generating manual annotation data of a (mobile eye-tracking) recording can be seen as a judgement task. Questions to be answered may include: ‘Is the subject looking at that particular object?’, ‘Is the subject looking at the left hand of the speaker?’, ‘Does the speaker make a gesture?’,‘Is that gesture then located at the center-center region of the gesture space?’, etc. In particular, in these repetitive tasks such as annotating gaze information, there is a high

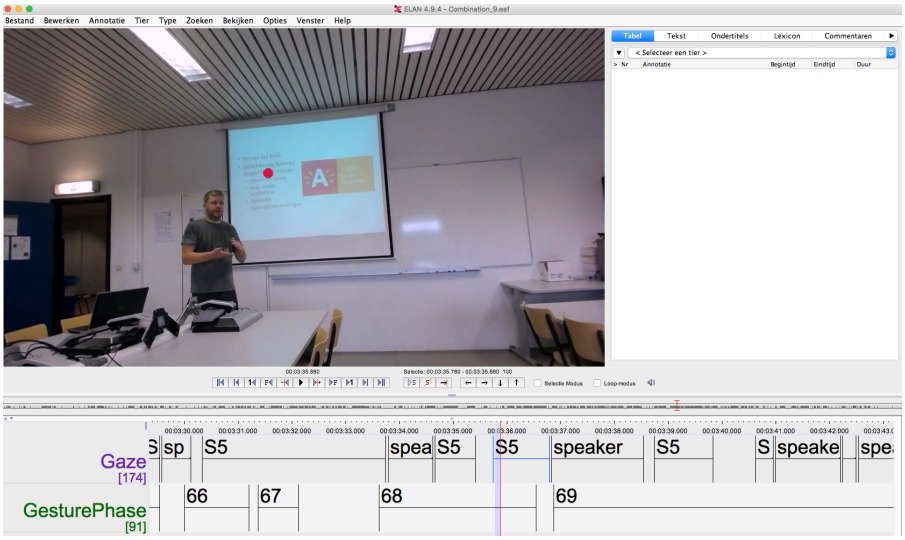


Figure 7.1: Screenshot of the ELAN annotation software in which a gaze and gesture tier of an eye-tracking recording are shown.

risk of mistakes, usually due to distraction or loss of attention. Especially when the annotation data are used for further analysis, it is of vital importance to validate the quality of the obtained data. In other words, the reliability of the data needs to be approved. It is important to verify that other annotators can agree with annotation values as assigned by the initial annotator. This explains why traditionally, multiple annotators are involved in the analysis of a single recording. The process of evaluating the reliability of data that is generated by multiple annotators is often referred to as *intercoder reliability*.

The more observers agree on the same observations, and the larger amount of data they scrutinise, the more one can assure that the data are reliable and that it can be exchanged with a clear conscience. It is clear that one needs a measurement to figure out this level of agreement. Choosing such an index is complex, since many reliability indexes are proposed in the literature. For example, Popping [106] compared 43 measures of nominal data. Unfortunately, some of these indexes respond to properties in the data that are not related to reliability at all. To avoid further confusion in the choice of a reliability index, a set of criteria for a good measure of reliability are given by Hayes and Krippendorff [60]. According to them, a good index of reliability should have the following properties:

1. It should measure the level of agreement between two or more annotators who performed the analysis task separately from each other. The reliability index may not be influenced by the number of annotators, nor their permutation.

2. The index should not be confounded by the number of categories or scale points that are available for coding. This will assure that the reliability index is not biased by the difference between the actual data and what the annotators imagine the data may be like.

3. The index must consist of a numeric scale of at least two points, that can not be interpreted erroneously. Perfect agreement should correspond to 100% and the absence of agreement corresponds to 0%. It is important that the reliability index does not overestimate the level of agreement.

4. A good reliability index should be able to handle any level of measurement: metric, nominal, ratio, ordinal, interval, etc.

5. The sampling behaviour should be known or at least computable, avoiding the need for estimations.

Keeping these criteria in mind, we give a brief overview of the most common reliability indexes and discuss their fulfilment of the above mentioned criteria.

Percent agreement

A well-known, and easy to understand reliability index is the *percent agreement*. It measures the proportion of units upon which two annotators agree. The formula of percent agreement A is given in equation 7.1 where O stands for the observed agreement and P for the possible agreement.

$$A = \frac{O}{P} \tag{7.1}$$

As discussed by Neuendorf [101], a percentage of agreement higher than 90% is always acceptable, and a percentage larger than 80% is acceptable in most cases. However, although the percentage of agreement is easy to calculate, it violates other criteria. It is only applicable for measuring the reliability between two annotators. Calculating the agreement becomes more difficult if more categories are taken into account, therefore, criterion 2 is violated. Furthermore, the percentage of agreement does not take into account the chance that an annotator made random guesses, making the achieved agreement in that case overestimated and therefore meaningless. To overcome this uncertainty, Scott's Pi and Cohen's Kappa were developed.

Scott's Pi

Compared to the percent agreement, Scott's Pi (π) [119] takes into account the possibility that a given value was annotated by chance. As shown in equation 7.2, it takes into account the expected agreement $Pr(e)$ next to the observed agreement $Pr(a)$. The expected percent agreement for the dimension is the sum of the squared proportions over all categories.

$$\pi = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{7.2}$$

Scott's Pi is only applicable for two annotators and nominal data and is therefore not the best choice for complex reliability measurements.

Cohen's Kappa

Cohen's Kappa [30] is calculated in the same way as Scott's Pi (see equation 7.2), however, it differs in terms of how $Pr(e)$ is calculated. Here, $Pr(e)$ is the hypothetical probability of chance agreement. This is calculated using the probabilities that each observer would randomly choose each category. Cohen [30] suggests to interpret the Kappa score as follows: 41-60% as moderate agreement, 60-80% as substantial agreement, and finally 81-100% as almost perfect agreement. Similar to Scott's Pi, Cohen's Kappa is suitable for only two annotators and nominal data.

Krippendorff's Alpha

The most reliable measure of agreement, although also the most complex and computationally difficult, is the Krippendorff's Alpha (α) [82]. In contrast to Scott's Pi and Cohen's Kappa, Krippendorff's Alpha measures the level of disagreement as shown in equation 7.3. D_o stands for the observed disagreement and D_e is the expected disagreement based on an interpretation of chance. The exact calculation of both disagreement values is rather complex and falls out of the scope of this dissertation. The reader can find more information regarding the α calculation in [81, 82].

$$\alpha = 1 - \frac{D_o}{D_e} \tag{7.3}$$

α satisfies each of the above-mentioned criteria and is therefore often proposed as the standard or best suited reliability measure. Regarding criterion (1),

the calculation of α is unaffected by the number of annotators, nor by their permutation. According to (2), α is exclusively extracted in the data that are generated by all observers. α defines a scale that ranges from 0% for the absence of reliability up to 100% for perfect reliability, and thus satisfies criterion (3). Regarding (4), α is applicable to both metric, nominal, ratio, ordinal or interval data. Finally, calculating α can be done without the need for any approximations. Furthermore, α can cope with incomplete or missing data. They achieve this by using a bootstrapping mechanism in which missing values are replaced by existing values from the dataset itself. Concerning the score of α, Krippendorf [82] suggests to require a minimum of 80%.

For the validation of our semi-automatic analysis approach, we use the same methodology that is described above. Therefore, we use our analysis framework for the automatic creation and annotation of the segments (as illustrated in figure 7.1). In a next step, we remove the annotation values of each segment and we ask an independent annotator to assign a label to each segment. Finally, we calculate the reliability between our automatically generated labels and the manual labels. Note that such a methodology only allows for a partial validation. The segments that are not detected by our analysis framework are ignored since we ask a human annotator to assign an annotation value only to the segments that are detected. However, we presented the results of a frame-based validation of each aspect of our framework at the end of each individual chapter. These validations do take into account the false negative detections and thus give a clear insight into the accuracy that we achieve. The purpose of these large scale experiments is to get insights in the applicability of our approach in the analysis of real-life and long-lasting recordings. In the following experiments, we report the four reliability measurements as discussed in this section (percent agreement, Scott's Pi, Cohen's Kappa and Krippendorff's Alpha).

7.3 Analysis of customer journey experiment

As mentioned in chapter 2, one of our experiments involved a large scale customer journey experiment in Museum M in Leuven (Belgium). The purpose of this experiment, in which 14 subjects participated, was to gain insights into the general experience of museum visitors. To obtain this information, we equipped each participant with a mobile eye-tracker before entering the museum. Then, they were instructed to buy a ticket for the Hieronymus Cock exhibition at the ticket counter. Next, they had to find their way to that exhibition and spend some time there. After approximately 30 minutes, each participant was instructed to return to the entrance of the museum, where the recording was terminated.

The idea behind this experiment was that these mobile eye-tracking recordings could provide a unique insight into the experience of the museum visitors. Together with a user experience bureau (Monkeyshot) and staff of the museum, we composed a set of research questions inquiring into the efficiency of the signage, the visibility of the walking guides that were available at the start of the exhibition, etc. An overview of the 7 questions is given below:

1. How much time did the participant spend at the ticket counter?
2. Did the participant look at the work of art in entrance hall?
3. Did the participant use the elevator or stairs to enter the exhibition?
4. Did the participant look at walking guides at the start of exhibition?
5. Did the participant notice the iPad at the end of the exhibition?
6. How much time did the participant spend at the exhibition?
7. Did the participant use the elevator or the stairs to get back from the exhibition?

We used our analysis framework to extract the relevant information from the eye-tracking recordings. More specifically, our object recognition approach was used for this task. As explained in chapter 3, our approach works as follows. While replaying an eye-tracking recording in our user interface, the annotator can select *objects of interest*, i.e. relevant objects in the recording. In a next step, our software counts when and how long a subject was looking at these relevant objects. Some examples of the selected objects are shown in figure 7.2. From left to right we distinguish: (a) an image of the ticket counter, (b) an image of the work of art that was located at the entrance hall of the museum, (c) an image of the control panel of the elevator and (d) an image of the walking guide shelf.

Using this approach, we automatically analysed the recordings of four participants. The remaining recordings suffered from some technical issues, such as inaccurate gaze detection, difficulties with the dark ambient light conditions inside the museum or difficulties with the batteries and were therefore prematurely terminated. After our initial analysis, answering questions such as *did the participant notice a specific object*, are indeed straightforward. To find out how much time each participant spent at the ticket counter, we let our software calculate the time between the first and last frame in which the ticket counter was visible within each recording. For the sake of completeness, we should mention that in the final analysis, we used multiple instances of the

(a) (b) (c) (d)

Figure 7.2: Examples of selected objects of interest in the context of the museum experiment.

Table 7.1: Questions to be answered in the context of the museum visit.

Question	Visitor 1	Visitor 2	Visitor 3	Visitor 4
1	1m22s	42s	49s	20s
2	NO	NO	NO	YES
3	Elevator	Stairs	Elevator	Stairs
4	1m43s	5m3s	1m21s	3m17s
5	NO	YES	YES	NO
6	NO	NO	YES	YES
7	28m58s	51m13s	35m3s	37m27s
8	Elevator	Stairs	Elevator	Stairs

same object. Indeed, we made use of our semi-automatic approach to improve the detection results.

In table 7.1 an overview of this analysis is given. For each of the four participants, an answer was given to each question, only relying on the semi-automatic analysis of the eye-tracking recordings. To cross-validate these results, we manually inspected each video and compared our results to a manual analysis. This comparison revealed that each question was answered correctly using our analysis framework. The only exception were the questions regarding the time spent on the ticket counter and on the entire exhibition, where we noticed some deviations.

In an additional analysis, we employed our person detection approach to quantify the number of human-human interactions throughout the recordings. Similar to the above-mentioned analysis, we selected a set of *objects of interest*. In this experiment, however, only two objects were selected: the route map and a specific work of art. We analysed three recordings using the object recognition

and person detection approach in our analysis framework. In contrast to the above-mentioned analysis, of which the results are presented in a table, we opted to present the results of this analysis in a time-based manner using a timeline representation. In figure 7.3 an illustration of such a timeline is given. A timeline visualises when and for how long a visitor looked at a specific object or person throughout the entire recording. This timeline obviously reveals that each visitor started his visit at the ticket counter, as shown by the detections of the desk attendant. Furthermore, this visualisation reveals that two visitors looked at their route map immediately after buying their ticket.

Manually analysing such a recording to get insight in how much time a subject spent at the ticket counter is rather straightforward since an annotator knowns in advance that this event occurs only at the start of the recording. On the other hand, manually counting how often and for how long the subject looked at his route map during the entire visit, is much more challenging since one needs to analyse each individual frame of the recording. It is clear that the manual analysis of the above-mentioned recordings, which contain approximately 58000 frames each, would take a substantial amount of time, whereas our approach only took a few mouse clicks for the selection of the objects of interest. The remainder of the analysis time was spent on the automatic processing and involved no manual intervention except for the selection of additional sample images, which, again, only took a few mouse clicks. Thus, although the automatic analysis of this experiment is rather confined in scope, it shows the full potential of our approach.

7.4 Analysis of a triadic conversation

A second, large-scale experiment in which we used our analysis framework was conducted in the context of a human-human interaction study. In this recording, consisting of a triadic set-up, the three participants were equipped with a Pupil-Pro mobile eye-tracker while they were involved in a natural conversation. Our analysis framework was used for the automatic analysis of the visual behaviour of one participant whose eye-trackers data will be automatically analysed in this experiment. In this case, the specific challenge for the analysis framework resides in the recognition of persons, and the automatic recognition of specific persons based on their clothes. Moreover, it is important to recapitulate that we process images that were captured by a mobile eye-tracker, thus we need to cope with moving camera viewpoint, motion blur, etc. This test will allow for a first insight into the system's reliability for the analysis of preferential looking in communication (e.g. a speaker looking at an audience).

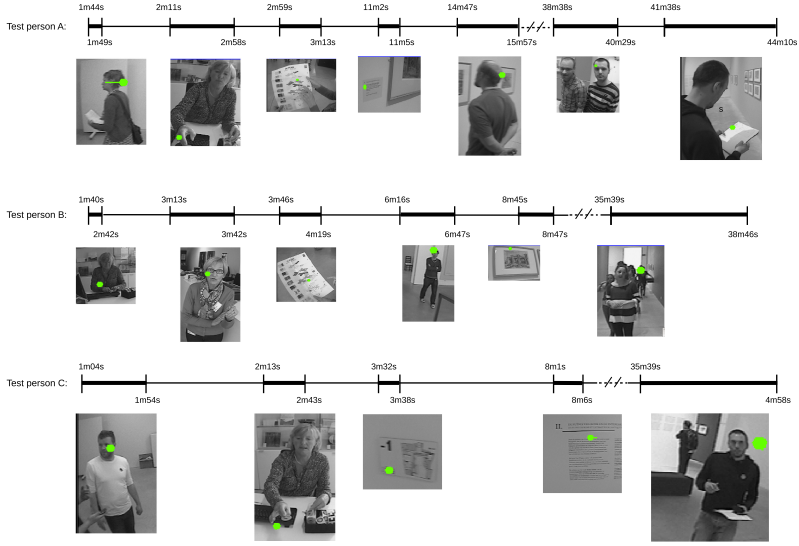


Figure 7.3: Results of our algorithm applied to the recordings of the museum visit. Each timeline represents a short summary of viewing behaviour of a participant.

This recording has a duration of 14m 17s and consists of 20568 frames. Initially, we were interested in the visual behaviour towards the two interlocutors. For this purpose, we used our person re-identification step as proposed in section 4.5. During a manual inspection of the video, we noticed that this participant tended to both look at a camera tripod and a poster on the wall while he spoke. Therefore, we also applied our object recognition software using two images of the objects of interest. In figure 7.4, an image frame of the mobile eye-tracker of the scrutinised participant is shown. In this image, we highlighted the four relevant items/persons: speaker 1, speaker 2, poster and camera tripod.

Again, this reveals the full potential of our approach. As compared to a marker-based analysis approach, in which each AOA object should be defined in advance, our approach allows researchers to investigate the visual behaviour towards any item that is present in the recording. Furthermore, it is even possible to reuse selected objects of interest in the analysis of other recordings.

As explained in section 7.1, we map the gaze data onto the detection result and export that result into an annotation file. A screenshot of the resulting file, in this case in ELAN format, is given in figure 7.5. The upper tier *Gaze_Auto* is the automatically generated annotation using our analysis framework. As

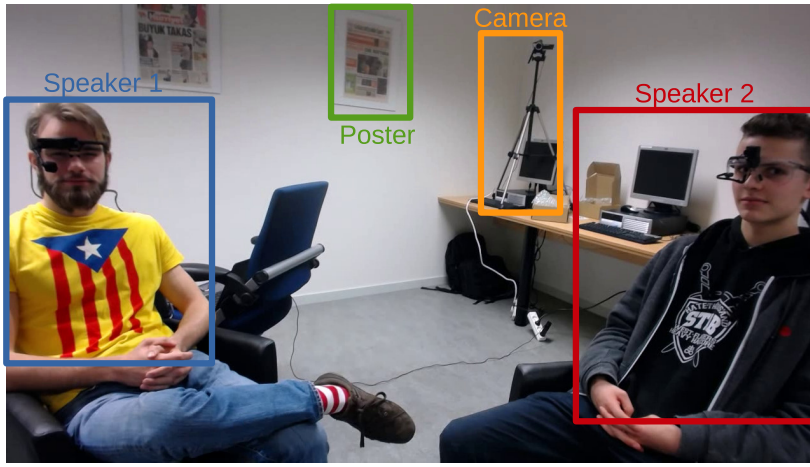


Figure 7.4: Different objects and persons that were automatically labelled using our software.

expected, four annotation labels were used, corresponding to the four relevant items as shown in figure 7.4.

This automatic analysis was validated thoroughly using the methodology that was described in section 7.2. Thus, for validation, we removed the labellings of each of the 463 segments and an independent annotator was instructed to manually assign a label to each segment, as shown in the lower tier of figure 7.5. The annotator could choose between the same four categories as our software did. In a next step, we calculated the agreement between the automatic and the manual labels. The result of this comparison is shown in table 7.2. This table reveals that the level of agreement between the manual and automatic analysis is very high (97,2%). On top of that, the stricter analysis methods such as Scott's Pi, Cohen's Kappa and Krippendorff's Alpha report very high levels of agreement. Based on these numeric results, we can conclude that our automatic analysis is suited for the analysis of this type of mobile eye-tracking experiments.

Although the accuracy of our automatic analysis approach is satisfactory, we looked into the disagreements between the manual and the automatic labellings. Manual inspection of the video revealed that most errors arise when the gaze cursor was positioned in between two objects of interest, as illustrated in figure 7.5. Here we see that the position of the gaze cursor is between the speaker and the camera tripod. This behaviour is indeed an artefact of an

Table 7.2: Reliability of triadic analysis.

	Level
Agreement	97.2%
Scott's Pi	96.0%
Cohen's Kappa	96.0%
Krippendorff's Alpha	96.0%

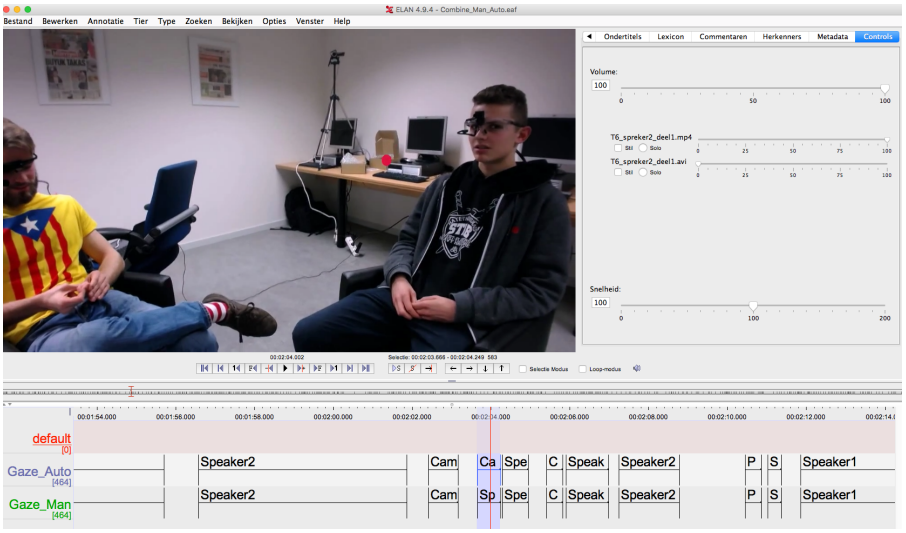


Figure 7.5: Example in which there is disagreement between manual and automatic annotation. The gaze cursor is indeed positioned between the speaker and the camera tripod.

automatic analysis approach. In manual analysis, one can interpret the situation and use additional information, such as speech, to make a deliberate choice. An algorithm, on the other hand, is unable to make such an interpretation and makes choices based on pure data.

Next to the accuracy, there is also a significant improvement in analysis time. The automatic analysis of the entire recording took approximately 27 minutes of computing time. In this analysis, both person detection and object recognition were performed simultaneously on a multi-threaded computing device. It is important to remind that in this analysis, the manual input was limited to selecting the two objects of interest and selecting a bounding box of each person for the histogram calculation. Indeed, this job can be done in less than one

minute of manual input. The remainder of the analysis task was performed fully automatic. As a comparison, the manual allocation of labellings to the segments in ELAN, which is only a part of the entire labelling job, took about 60 minutes. It is clear that manually annotating an entire recording is labour intensive.

7.5 Analysis of lecture recording

Finally, we performed the analysis of another recording which was, again, made in the context of a human-human interaction experiment. In this experiment, a participant was equipped with a mobile eye-tracker while attending a PowerPoint presentation given by a speaker. This recording had a duration of 4m 47s and consists of 6700 frames. In total, we performed three independent analyses on this recording. In the first one, as discussed in section 7.5.1, we were interested in the visual behaviour of the participant towards relevant body parts of the speaker and the presentation screen. The focus of this experiment lies on the visual behaviour towards face and hands. In the second analysis (7.5.2), we investigated the visual behaviour towards the speaker and each individual presentation slide. The purpose of this experiment was to test our object recognition approach to the limit, since the differences between the individual presentation slides were minimal. We deliberately chose to perform this gaze-based analysis on two different levels since it allows for a better insight in the accuracy performance of each part of our framework. Furthermore, for validation of this analysis, we removed the annotation values of the automatically created segments and a human annotator was instructed to manually assign a label to each segment. Each analysis was annotated by another annotator to avoid bias.

In the third and final analysis (7.5.3), we used our gesture segmentation for the analysis of the gestures that were made by the speaker.

7.5.1 Body parts versus presentation screen

As mentioned above, this analysis was performed to gain insights into the visual behaviour of a spectator towards relevant body parts of a speaker who is giving a PowerPoint presentation. This context includes several analytical challenges. Firstly, the speaker is mobile, thus he may walk in front of the presentation screen. Secondly, due to the spontaneous nature of these recordings, the speaker regularly turns towards the presentation screen, which makes the face detection difficult. Finally, the recording was made using our Pupil-Pro eye-trackers, which embed an eye-camera with a relatively low frame rate, as compared to

more advanced mobile eye-trackers. As a result, the eye-tracker sometimes fails to detect short fixations (150-200ms). In particular when analysing the visual behaviour towards rapid moving items, such as the hands of the speaker, this problem may emerge.

Our framework was used for the analysis of this recording. More specifically, we used our object recognition technique to detect when and for how long the subject was looking at the presentation screen. Therefore, we defined one object of interest that consists of several example frames of the presentation screen. These example frames include images of the presentation screen both with and without visual content. Furthermore, we used our person detection approach together with the face detection to detect how often and for how long the subject was looking at the speaker, and more specifically at his face. Finally, we used the semi-automatic segmentation-based hand detection approach to analyse the visual behaviour towards the hands of the speaker. As a final aspect of this automatic analysis, the gaze data is automatically mapped on each detected item in the recording. As previously explained, we artificially enlarge the upper body and facial bounding boxes in order to cope with some deviations of the gaze cursor and to ensure sufficient overlap between the detected body parts and the gaze cursor. In case of the hands, we allow a certain distance between the endpoint of each hand and the gaze cursor. In this case, the maximum distance equals the half-face width as used in the validation of the hand detection approach.Based on this mapping step, the segments were created and labelled.

These automatically generated annotations can then be used for further research, for instance for exploring the impact of multimodal cueing as a presentation technique (i.e. the use of gesture and eye gaze by a speaker to draw the audience's focus of attention to a specific object, person, or presentation slide). As our system provides both speaker information through face and gesture detection and audience information through the gaze coordinates, the mapping of both allows for a largely automatic analysis of the correlation between gesture and gaze behaviour [22, 59].

For the validation of this analysis, we removed the annotation values of a subset of the segments and a human annotator was instructed to manually assign a label to each of the 90 segments. From these 90 segments our automatic analysis labelled 59 segments as *face*, 27 as *presentation screen*, 2 as *upper body* and 2 as *left hand*. The annotator had to choose between the same annotation values used by our framework: face, upper body, left hand, right hand, presentation screen. The human annotator labelled 62 segments as *face*, 26 were labelled as *presentation screen*, 1 was labelled as *upper body* and 1 was labelled as *left hand*.

Table 7.3: Reliability of lecture analysis. Items under scrutiny: face, upper body, hands and presentation screen.

	Level
Agreement	95.6%
Scott's Pi	90.4%
Cohen's Kappa	90.4%
Krippendorff's Alpha	90.4%

The result of comparing the automatic versus the manual analysis is given in table 7.3. This table reveals, again, that our framework is capable of analysing this type of recordings in a highly accurate manner. In particular, the semi-automatic analysis of visual behaviour towards relevant body parts has proven to be highly accurate.

During the analysis of this recording, we noticed that the subject only sporadically looked at the hands of the speaker. This could, however, be caused by the limited accuracy and lower frame-rate of the Pupil-Pro eye-trackers. This does not jeopardise the accuracy performance of our approach as such, but researchers who use this data for further analysis should be aware of the fact that the system can only be as reliable as the recorded data on which it performs its analysis.

7.5.2 Speaker versus slides

In the second analysis of this recording, our main focus shifted from visual behaviour towards relevant body parts to the visual behaviour towards each individual slide that was shown during the PowerPoint presentation. During this analysis, we did take into account the visual attention towards the speaker. However, no distinction was made between looking at individual body parts. An image frame of each slide was selected manually while replaying the video. The selected images are illustrated in figure 7.6. Each slide was represented by only one example image. As seen in this figure, four of the six slides only contain textual information, which makes it hard to distinguish them in a detection step. In particular the difference between the third and fourth slide is minimal from the perspective of vision technique. Furthermore, the presentation screen is often occluded by the speaker himself. Thereby it might be difficult to disambiguate looking at the speaker and looking at the presentation screen.

Using our analysis framework, we processed the entire recording, resulting in 174 segments in which the subject was looking at one of the items under scrutiny.

Table 7.4: Reliability of lecture analysis. Items under scrutiny: speaker and each individual slide.

	Level
Agreement	94.8%
Scott's Pi	92.8%
Cohen's Kappa	92.8%
Krippendorff's Alpha	92.8%

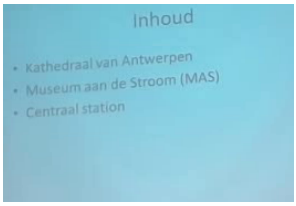
From these 174 segments our automatic analysis labelled, 7 segments as *slide 1*, 3 as *slide 2*, 24 as *slide 3*, 17 as *slide 4*, 25 as *slide 5*, 18 as *slide 6* and finally 80 segments as *speaker*. Similar to the validation of the previous experiments, we removed the annotation labels of each segment and an independent annotator was instructed to manually re-label them. The human annotator labelled 7 segments as *slide 1*, 2 as *slide 2*, 21 as *slide 3*, 18 as *slide 4*, 22 as *slide 5*, 18 as *slide 6* and finally 86 segments as *speaker*. Then, the level of agreement between both annotation files was measured, as shown in table 7.4. Again, this reliability measurement reveals that our approach is applicable to the analysis of this type of recordings. Even when several objects of interest are highly similar, our approach is able to analyse the visual behaviour in a precise manner.

The duration of our automatic object recognition analysis for this recording was only 14 minutes. The person detection, on the other hand, took 9 minutes 46 sec. However, on a modern multi-threaded computer, both approaches can run simultaneously. Again, it is important to remember that in our automatic analysis approach, the manual work is restricted to only selecting the objects of interest, making our approach less labour intensive as compared to fully manual analysis. The remainder of the analysis time is entirely spent by the computer.

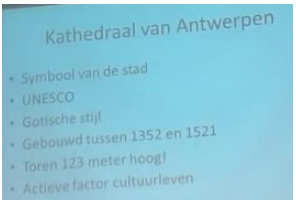
7.5.3 Gesture analysis

For a final analysis, we applied our gesture segmentation approach to this recording to identify the gestures that were made by the speaker. Once the gestures were segmented, we mapped the gaze data on top of the respective hand positions to gain insights into the visual behaviour towards these gestures. As already mentioned in section 7.5.1, the Pupil-Pro eye-tracker has difficulties in detecting short fixations. Therefore, the relationship between *gestures made by the speaker* and *visual attention of the participant to these gestures* might not be representative. Nevertheless, mapping the available gaze data on top of these detected gesture sequences is useful to prove the full potential of our approach.

(a) Example frame of slide 1

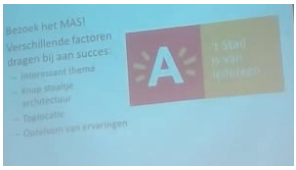


(b) Example frame of slide 2



(c) Example frame of slide 3

(d) Example frame of slide 4



(e) Example frame of slide 5

(f) Example frame of slide 6

Figure 7.6: Selected objects of interest for the analysis of the lecture recording.

As mentioned in chapter 6, our gesture analysis builds on our semi-automatic, segmentation-based, hand detection approach. During the detection of the hands of the speaker, our system requested manual input in 120 frames, which is only 1.7% of the 6700 frames. This analysis, including the generation of the candidates as well as filtering and manual interventions, took approximately 29 min, of which only a fraction was spent on manual analysis. Based on the retrieved information (i.e. face, upper body and hand locations), our gesture segmentation approach identifies the segments in which the speaker is gesturing. In total, 91 gesture sequences were found. Processing the detection file that contains the information of the relevant body parts and generating the gesture sequences took approximately 4 minutes.

For validation of our gesture segmentation approach, we asked an independent annotator to manually assign a label to each extracted segment. The annotator could choose between either *gesture* or *non-gesture*. Then, we compared the automatically generated annotations to the manually assigned labels to measure the reliability of our gesture segmentation approach. As shown in the leftmost columns of table 7.5, the reliability of our approach is again satisfying, although the overall score is somewhat lower as compared to measurements of the previous experiments.

Manual inspection of this result revealed that the majority of disagreements occur in short gesture sequences. Indeed, it might happen that the Kalman filter

Table 7.5: Reliability of gesture analysis.

	Level	**Level without short segments**
Agreement	79.1%	87.5%
Scott's Pi	78.7%	87.3%
Cohen's Kappa	78.7%	87.3%
Krippendorff's Alpha	78.7%	87.4%

of at least one hand floats away before our hand detection approach requests manual intervention, resulting in false positives. Furthermore, as shown in figure 7.7, there are some translations in the upper body detections. These are mainly caused by the deformable aspect of the model that we use. Since the relative hand positions are calculated using the center of the upper body detection, their position is affected by these translations. As a result, it might happen that the distance between the rest position and the relative position of some hands exceeds the threshold, causing an erroneous gesture sequence. Besides these translations, we also noticed slight scale variations in the upper body detections, which also affect the obtained hand positions.

As an additional validation step, we removed the gesture sequences which have a duration less than 500 ms, which is indeed relatively short for a gesture, and we repeated the reliability measurements. The rightmost column of table 7.5 shows the improved results of this additional validation. It is clear that the lower reliability scores in the left part of this table are indeed mainly caused by the shorter gesture segments.

The final step in this analysis involved the automatic mapping of the (available) gaze data on top of the detected gesture sequences to get a first, rough, insight into the visual attention towards the gestures of the speaker. Our framework automatically counts how often and when the participant looked at the gestures of the speaker. In this particular recording, the participant looked 10 times at the hands of the speaker. Nine times out of the ten, the participant looked at the hands while the speaker was gesturing. Thus, in only one case, the participant looked at the hands of the speaker while they were in rest position. An example frame in which the participant looked at a gesture of the speaker is given in figure 7.8.

This section demonstrated the applicability of our approach for the analysis of various types of eye-tracking recordings. Based on the reliability measurements, we proved the accuracy of our approach for the analysis of visual behaviour towards specific objects, relevant body parts and gestures. The ability to automatically generate such annotations is a major step ahead in the analysis

(a) (b)

Figure 7.7: Variations in upper body detections that may cause changes in relative hand positions.

of this type of recordings. It allows researchers to spend their time on their own research questions rather than spending time on the necessary, but labour-intensive, (initial) annotation of the video data.

7.6 Visualization of data

As already mentioned, we developed a tool that transforms the results of our framework into an XML-compatible file. This makes our approach integrable with existing annotations that were often created using annotation tools such as ELAN or ANVIL. Whereas this output format is commonly used in the analysis of e.g. human-human interaction experiments, it is rather inapplicable in the analysis of customer journey or market research experiments. To further enlarge the applicability of our approach in these application domains, we developed a series of visualisation methods for representing the analysis of an eye-tracking experiment.

In one of the master theses that I supervised (Cleymans [29]), we developed a user-friendly visualisation environment for the analysis of mobile eye-tracking

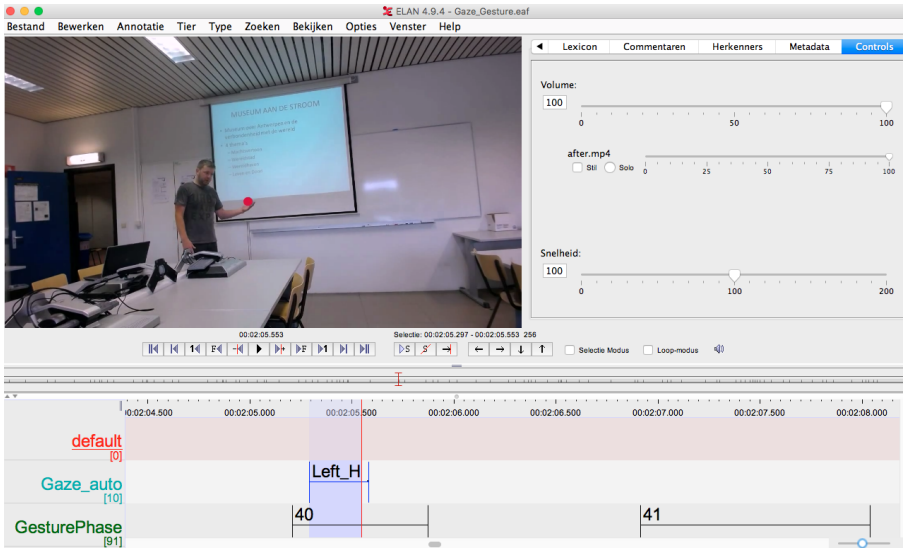


Figure 7.8: Illustration of a participant who is looking at a gesture that is made by the speaker.

experiments. The purpose of this tool is to represent the raw, frame-based results that were generated using our framework, in an attractive and informative manner. Within this tool, several data representation methods were implemented as illustrated in figures 7.9 and 7.10. In this figure, we visualise the analysis of a subset of one of the recordings that was made during the customer journey experiment in Museum M. In this analysis, we selected 4 relevant objects of interest. Furthermore, the face and upper body detection was used to count how often and for how long the subject looked at another person.

7.6.1 Visualisation of numerical data

Figure 7.9(a) shows a first visualisation method, viz. representation of the raw numerical data of how often and for how long the subject looked at a particular item or object. Such numerical data are relevant for interpreting eye-tracking recordings in the context of marketing experiments, in which researchers are interested in which brand attracted the most visual attention. In figure 7.9(b), these numerical data are represented as bar graphs that visualise which object or item attracted the most visual attention. Furthermore, each object of interest

is represented by its own unique colour, which makes the interpretation of the results easier. The same colours are used throughout the other visualisation methods as well. Figure 7.9(c), shows a similar visualisation method i.e. the numerical data are represented as a pie-chart.

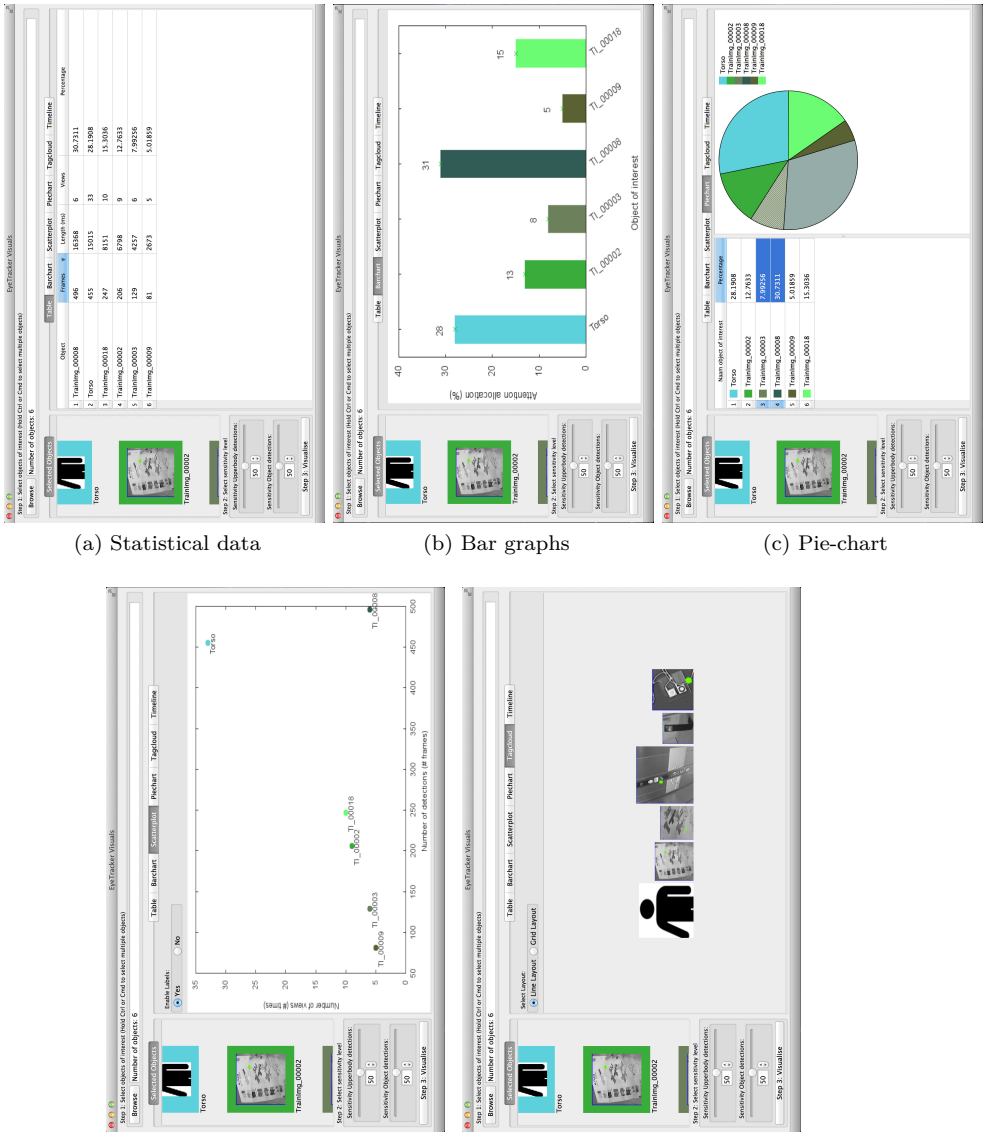
Another method of representing the data is found in figure 7.9(d). Here, a scatter plot visualisation is used to represent the relationship between the total view time and the number of times that the subject looked at a given object or item. Such a graph directly reveals whether an object attracted one long visual fixation or multiple shorter visual fixations. In this particular example, we see that the subject looked at the face of another person 33 times, in a total of 455 frames. On the other hand, the subject looked at one of the objects of interest for a total of 496 frames, but only 6 times. This case makes it clear that the scatter plot visualisation is useful in the interpretation of the analysis data.

In figure 7.9(e), the data is represented using an object cloud. This visualisation method is inspired by the well-known tag clouds in which a text is visually represented by the most frequent or most important words. The importance of each word is then shown by its font size. The same methodology is used in our object cloud: object or items that attracted more visual attention are represented at a larger scale.

It is important to note that, besides statistics on the number and length of fixations, our framework could be used for retrieving other relevant eye-tracking measurements. For example, one could easily extract the time until the first fixation, or the total view time w.r.t. a given object as expressed in percentage of the total recording, the number of fixations per minute, etc.

7.6.2 Timeline visualisation

The above-mentioned visualisations are suited for representing the analysis of an experiment in which the temporal aspect is irrelevant. However, if the sequence in which a subject looked towards the relevant object is important, another visualisation method is required. To realise this, we developed a representation method, in which each visual fixation is chronologically displayed on a timeline. Such a visualisation is useful for the analysis of customer journey experiments in which researchers are interested in the trajectory of the subjects. In figure 7.10(a), an illustration of our timeline representation is given. Here, we see the objects or items that were viewed by the subject in chronological order. For each visual fixation, we mention start and end time. In long-lasting experiments, it might be difficult to maintain the overview in this type of data representation. Therefore, we developed another visualisation method, in which the entire timeline is represented within the width of the user interface, as



(a) Statistical data (b) Bar graphs (c) Pie-chart

(d) Scatter plot (e) Object cloud

Figure 7.9: Visualisation methods for representing the numerical analysis data of an eye-tracking experiment.

shown in figure 7.10(b). This representation allows researchers to get a clear overview of the chronological order in which the subject looked at the relevant objects or items.

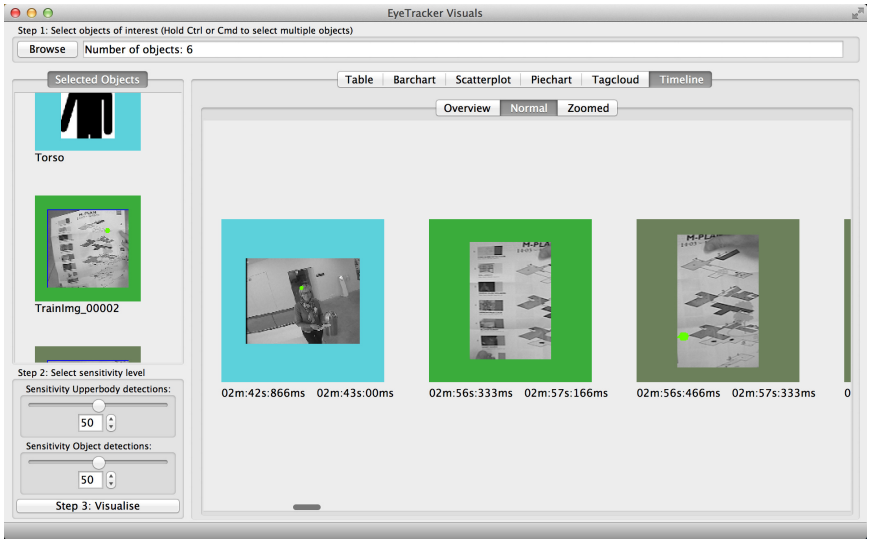
7.6.3 Heat map visualisation

Heat map visualisations are well known in the context of eye-tracking. Traditionally, they are used for the visualisation of screen-based eye-tracking experiments in which they display the spatial distribution of visual attention in e.g. usability research for websites. A heat map is generally generated referenced to a fixed window. Applied to mobile eye-tracking, in which the camera viewpoint (i.e. the scene camera of the mobile eye-tracker) can move freely, we do not have such a fixed reference frame, making the creation of such a heat map far more complex.

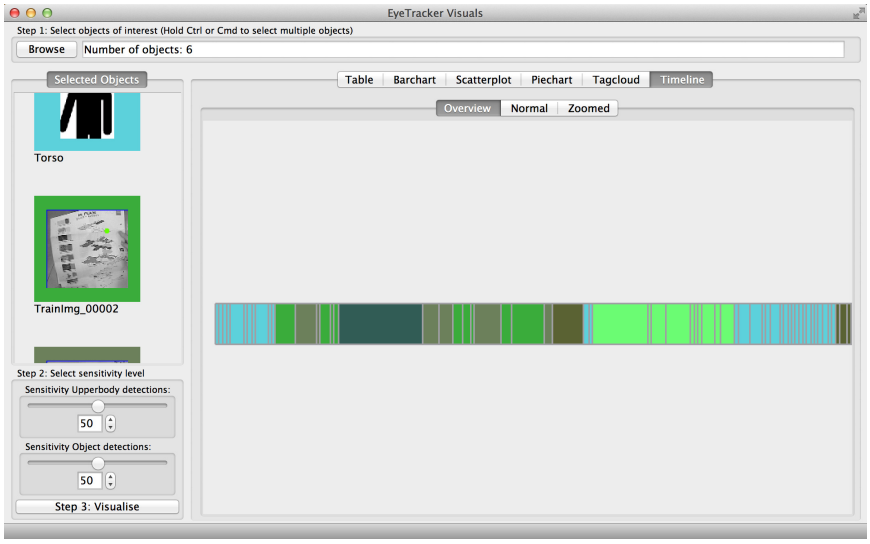
Nevertheless, we managed to create a heat map visualisation of a mobile eye-tracking experiment. To do so, we chose to visualise the visual attention towards relevant body parts such as hands or faces in the context of a human-human interaction experiment. To overcome the issue of the missing reference frame, we use the hand and head detection in each frame to transform the gaze position to a standard pose of a person using an affine geometrical transformation. The resulting heat map is shown in figure 7.11. This figure reveals that the left hand attracted the most visual attention, indeed the numerical data of this experiment indicates that the left hand attracted 30.2% of the visual attention, whereas only 17.3% of the visual fixations were positioned at the right hand.

7.7 Conclusion

This chapter served as a final validation of our semi-automatic analysis framework. Throughout this chapter, we analysed various eye-tracking experiments using our framework. This analysis includes the detection of various objects or body-parts in the images that were captured by the scene camera of a mobile eye-tracker, as well as mapping the gaze data on top of these detections. This allows us to automatically count how often and how long a subject spent visual attention towards the relevant objects or items. To verify the reliability of our automatically generated annotations, the recordings were manually annotated by independent annotators and compared against our automatic labellings. However the used methodology does not take into account the false negative detections (i.e. the segments that were missed by our framework), it does provide an meaningful validation of the proposed framework when



(a)



(b)

Figure 7.10: Automatically generated timelines of an entire experiment.

Figure 7.11: Heat map of an eye-tracker experiment.

applied for the analysis of long-lasting and real-life recordings. Furthermore, each part of our analysis framework was validated using a frame-based method as described at the end of each chapter. In these frame-based validation experiments, where the false negatives were taken into account, we also report high accuracy. For comparing the automatically generated annotations and the manual labellings, we calculated the level of agreement between the annotations as well as more complex reliability measurements such as the Krippendorf's Alpha. These comparisons revealed that our object recognition and person detection approaches achieve very high reliability, making them indeed applicable for the analysis of real-life mobile eye-tracking experiments. The reliability of our gesture segmentation is more sensitive and therefore achieves a slightly lower reliability. Nevertheless, the achieved reliability remains very high and proves that the gesture segmentation technique is indeed applicable for the analysis of real-life recordings.

Besides validating our approach, we presented various visualisation methods that represent the output of our framework in a visual way. These visualisations contribute to the applicability of our framework by improving the interpretability of the obtained data. Besides transforming our output to an XML-based file, which is compatible with annotation tools such as ELAN, we provide additional visualisations. These include visual representations of the statistical data, timeline visualisation as well as the well-known heat map representations.

Chapter 8

Conclusion and Future work

8.1 Conclusion

The main goal of this dissertation was the development of an automatic framework for the efficient and accurate analysis of mobile eye-tracking recordings. Such an analysis consists of annotating the visual behaviour of a subject (i.e. the person wearing a mobile eye-tracker) towards relevant items. Depending on the purpose of the recording, these relevant items vary from specific products in a market research experiment up to the face of a speaker in a human-human interaction experiment. Traditionally, video recordings are annotated manually, which is a labour-intensive, time-consuming and error-prone task. Several commercial systems for the analysis of mobile eye-tracking recordings exist (marker-based analysis or software-based analysis such as Tobii Pro Glasses Analyzer). However, either they restrict the flexibility of mobile eye-tracking recordings to lab conditions or they are only applicable in a limited number of real-life applications. Since the labour-intensive analysis of these recordings is indeed one of the reasons why mobile eye-tracking is often ignored in research experiments, the development of an automatic analysis framework will broaden the applicability of mobile eye-tracking within various application domains.

Developing a framework for the analysis of real-life mobile eye-tracking recordings, that is both highly accurate and efficient, is challenging. Besides the challenges directly related to mobile eye-tracking (i.e. moving camera viewpoint), we also encounter (fast) moving objects in the scene resulting in motion blur. Moreover, the size of relevant objects in this type of images is

often very small, making them hard to detect and track over time.

In our proposed framework, the focus lies on the detection of relevant objects as well as the detection of various body parts. Due to our object recognition approach, our system is capable of measuring the visual behaviour of a subject towards any object or item that is visible in the images that were captured by the scene camera (i.e. the forward looking camera of a mobile eye-tracker). Compared to the marker-based analysis, our approach no longer requires that relevant objects are defined in advance. Thereby, our object recognition approach is better suited for the analysis of unrestricted real-life experiments. Furthermore, automatically mapping the gaze data on detected body parts such as hands, faces and upper bodies in the images captured by the scene camera, is a significant step forwards in the analysis of human-human interaction experiments. Finally, our gesture analysis tool, which builds on the body-part detection, allows us to automatically measure the relationship between visual behaviour and gestures that are performed by an interlocutor.

The main topics that were addressed in this dissertation were chosen in such a way that they cover a wide range of mobile eye-tracking applications. Furthermore, since our approach solely relies on the images that were captured by the scene camera of an eye-tracker, our approach is unobtrusive and therefore applicable in any type of mobile eye-tracking experiment.

Besides reducing the manual workload that is related to this kind of analysis task, another vital aspect is to avoid any compromise on the accuracy of the analysis. Therefore, we selected the best suited computer vision algorithms and we developed a semi-automatic analysis approach, in which, and only when required, manual intervention is incorporated into the automatic analysis to further improve the accuracy. The ability of manual interventions ensures a certain level of control to the user, whereas fully automatic approaches are often black-box systems in which interpretation and/or correction of false detections is much more complicated. The concept of manually intervening in an automatic approach is not only relevant in this application, since the methodology can be generalised across various applications.

As presented in our last chapter, we thoroughly validated our developed framework in terms of efficiency and effectivity in the analysis of various real-life mobile eye-tracking recordings. By comparing our automatically generated analysis against manual analysis, we were able to measure the reliability of our approach. These experiments revealed that each aspect of our approach is able to achieve more than 80% on a strict reliability measure such as the Krippendorf's Alpha. Although this comparison does not take into account the sequences that were missed by our framework, it confirms the general effectivity of our framework. Furthermore, it is important to note that the entire analysis

of an eye-tracking recording is finished when passing through our framework. The main purpose of our framework is to reduce the manual workload as much as possible. However, the human annotator can always intervene and refine the obtained result when necessary. On top of the convincing accuracy results, these experiments revealed that the required analysis time of our framework is significantly shorter as compared to a fully manual analysis. Furthermore, only a tiny fraction of this analysis time involves manual coding, since the majority of the analysis is performed automatically. For example, in our object recognition approach the manual input is limited to the initial selection of the objects of interest and, only if required, some additional manual interventions. Our semi-automatic hand detection approach requires more manual interventions, nevertheless our experiments revealed that we reduce the amount of manual labour by a factor of 37 as compared to fully manually analysing each individual frame. In other words, although manual intervention is intertwined in our analysis framework, the amount of manual labour is only a fraction compared to fully manual analysis.

Finally, it is important to mention that we provide a range of output formats making our automatically generated annotations integratable with existing annotations and easily interpretable by researchers.

At last, we would like to mention that our analysis software will be made publicly available on `www.eavise.be/insightout`. Hence, other researchers can use our framework to reduce both the analysis cost and time that is related to mobile eye-tracking recordings. Hopefully, our framework will somehow contribute to the increasing popularity of mobile eye-tracking.

8.2 Future work

Although our analysis framework has proven to be valuable in the analysis of mobile eye-tracking recordings, there is room for improvement and future development. We start this section by giving an overview of straightforward improvements as well as some more advanced enhancements that could further improve the applicability of our approach. Furthermore, we give an overview of possible developments within the field of mobile eye-tracking and the analysis of the recorded data.

We noticed that the upper body detections are sometimes not perfectly aligned with the persons that are presented in the images. As a result, it might happen that some gesture segments are created erroneously. This issue can be solved by tracking the detection scale at which an upper body is detected. By doing so, one could limit the search space and therefore remove the fluctuations in

the size of the detection windows. Furthermore, a more advanced tracking may overcome the translation issues.

Another straightforward enhancement consists of expanding our hand and gesture detection approach to multiple persons. Currently, our approach is developed to detect the hands of only a single person. The ability to analyse the gestures of multiple persons is relevant in for example the analysis of multi-party interactions. Furthermore, integrating our person re-identification step in the hand and gesture detection approach is inevitably linked with this expansion. Such an integration will bring the analysis of e.g. triadic conversations to a next level by automatically analysing the gestures of each participant as well. Building on our gesture detection approach, a gesture recognition approach can be developed allowing for a fine-grained gesture analysis. The automatic recognition of several basic gesture patterns, such as pointing or batons, is relevant in various application domains including research on gestural behaviour and research on sign language for the automatic subtitling of singers. Finally, our gesture analysis approach could be expanded as well. Currently we define a single rest position for each hand, but evidently it might occur that the rest position of the speaker changes during the experiment. Therefore, our approach can be modified by searching for multiple rest positions, based on the density of the relative hand positions. Another modification that could improve the accuracy of the gesture segmentation approach is combining the displacement to the resting position with the velocity of the hands. Such an integration would allow for a more accurate and finer segmentation, and therefore an accurate identification of a hold phase in a pointing gesture would be possible. Furthermore, it would be interesting to measure the influence of multiple annotators on the manual input of our semi-automatic analysis. We do expect that our system is highly robust since manual intervention is only required sporadically. Therefore, the manual annotation is no longer a repetitive task, reducing the chance of erroneous manual annotations.

A more advanced modification can be made in the part of the object recognition. Currently, our framework requires a manual selection of objects that are interesting in a recording. Instead of manually selecting objects of interests, one could develop a system that automatically returns the relevant objects a subject was looking at. Thus, for each visual fixation, one could store the respective object and thereby creating a collection of objects that were viewed by the subject. Based on that collection one could extract statistics of the viewed objects in terms of occurrences or total viewing time.

Based on the trends that emerge in the field of computer vision, we believe that, probably within a few years, technology will exist to expand this analysis framework in two ways. On the one hand, both the algorithms and necessary computational power will be available to perform the entire analysis in real-time.

In that case, a human annotator could then perform the needed interceptions on the fly. On the other hand, the development of powerful algorithms such as CNNs are booming. Compared to the techniques that were used in our approach (i.e. object recognition and object detection), these algorithms are capable of detecting multiple object classes, allowing for automatic scene understanding. This type of analysis enables a new level of analysis. For example, based on the appearance of several objects (i.e. sofa, refrigerator, etc.), these algorithms can distinguish a living room from a kitchen. Especially in experiments where the localisation of the subjects is important, these algorithms may provide useful information. Besides using these algorithms for localisation purposes, one could also expand the proposed framework by developing another method of analysis. Instead of only analysing the smaller region around the gaze cursor, one could detect relevant items in each entire image that is captured by the scene camera. By then validating the visual behaviour w.r.t. these relevant items, one could make the distinction between exposure and attention, which is indeed a relevant measure in various eye-tracking experiments.

Besides these analysis methods, techniques such as monocular simultaneous localisation and mapping (SLAM) allow for the automatic creation of a 3D model of the trajectory of the participant. By combining this automatic localisation and automatic analysis methods, as the ones presented in this dissertation, one automatically gets insight in which object attracted the most visual attention as well as where in the trajectory that object was viewed. That may result in 3D heat maps in which the positions of the relevant or important objects are highlighted. This could be particularly useful in the analysis of customer journey experiments.

Another interesting application domain can be found in the combination of mobile eye-tracking and egocentric vision. Today's society is all about visual communication and sharing personal information and experiences through video logging (vlogging). Hence, in recent years, there is a growing interest in the analysis of egocentric videos. In that research field, algorithms are developed for the automatic summarising the recorded activities. By integrating mobile eye-tracking into these recordings, one can further refine the summarisation by automatically highlighting important objects based on the amount of visual attention that was given to them.

Ultimately, mobile eye-tracking should be able to differentiate between attentively looking on the one hand and daydreaming on the other hand. By predicting the visual attention by solely relying on the images that are captured by the scene camera, one is one step closer to the ability to automatically measure the level of attentiveness. Crucial in such an approach is to predict the locations in the scene that might or should attract visual attention. Besides determine the salient regions in an image, such as proposed in the well-known

saliency model of IItti & Koch [65], other cues such as the gaze direction of others persons [104] are also relevant measurements for the prediction of visual behaviour. A survey of methodologies for the study of visual attention is given in [126]. By comparing the actual visual behaviour to the predicted behaviour, one can get insight in the attentiveness of the subject. Needless to say, this capability is relevant for a wide range of application domains, including marketing and healthcare.

Besides advances in computer vision techniques, we are convinced that the mobile eye-trackers themselves will evolve within the next years. In particular, the integration of a traditional mobile eye-tracker with a 3D-scene camera would be a useful development. Such a combination will pave the way for autonomous localisation during real-life mobile eye-tracking experiments.

However, to fully grasp the potential of mobile eye-tracking and our framework in particular, we initially need to focus on the important task of actually exerting our developed framework for the analysis of various mobile eye-tracking recordings. Subsequently, in September 2016, a research project has started in which our framework will be used by several marketing bureaus for the automatic analysis of their mobile eye-tracking recordings. Surely, this will be a first step towards a bright future in which the manual analysis of mobile eye-tracking data becomes redundant.

Bibliography

[1] Anatomy of the human eye. http://www.minbreak.com/amazing-10-diagram-basic-part-eye-anatomy-of-the-human/. 2016-08-03.

[2] AUVIS. https://tla.mpi.nl/projects_info/auvis/. 2016-10-06.

[3] Capture natural shopper behavior in the most cost-efficient way. http://acuity-ets.com/products_glasses.html. 2016-06-05.

[4] Dual purkinje eyetrackers. faculteit psychologie en pedagogische wetenschappen KU Leuven [online]. http://ppw.kuleuven.be/english/lep/resources/purkinje. 2012-04-11.

[5] Fields of use of Tobii eye-tracking: Marketing and consumer research. http://www.tobiipro.com/fields-of-use/marketing-consumer-research/advertising/. 2016-06-05.

[6] RED500 screen-based eye-tracker developed by SMI. http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/red250-red-500.html?gclid=CKO43dLnpM4CFUoz0wodBGMIDA. 2016-08-03.

[7] Scenecamera eye tracking by Arrington. http://www.arringtonresearch.com/scene.html. 2016-08-01.

[8] Eye movements during information processing tasks: Individual differences and cultural effects. *Vision Research 47*, 21 (2007), 2714 – 2726.

[9] Abuczki, Á., and Ghazaleh, E. B. An overview of multimodal corpora, annotation tools and schemes. *Argumentum 9* (2013), 86–98.

[10] Alon, J., Athitsos, V., Yuan, Q., and Sclaroff, S. A unified framework for gesture recognition and spatiotemporal gesture

segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 9 (2009), 1685–1699.

[11] Alves, R., Lim, V., Niforatos, E., Chen, M., Karapanos, E., and Nunes, N. Augmenting customer journey maps with quantitative empirical data: a case on eeg and eye tracking. In *Proceedings of Designing Interactive Systems* (2012).

[12] Badi, H. Recent methods in vision-based hand gesture recognition. *International Journal of Data Science and Analytics* (2016), 1–11.

[13] Baumberg, A. Reliable feature matching across widely separated views. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2000), pp. 774 –781 vol.1.

[14] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features (SURF). *Computer Vision and Image Understanding 110*, 3 (June 2008), 346–359.

[15] Benenson, R., Mathias, M., Tuytelaars, T., and Van Gool, L. Seeking the strongest rigid detector. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 3666–3673.

[16] Bennewitz, M., Axenbeck, T., Behnke, S., and Burgard, W. Robust recognition of complex gestures for natural human-robot interaction. In *Proceedings of the Workshop on Interactive Robot Learning at Robotics: Science and Systems Conference (RSS)* (2008).

[17] bin Abdul Rahman, N. A., Wei, K. C., and See, J. RGB-H-CbCr skin colour model for human face detection. *Faculty of Information Technology, Multimedia University* (2007).

[18] Bo, N. B., Dailey, M. N., and Uyyanonvara, B. Robust hand tracking in low-resolution video sequences. In *Proceedings of International Conference: Advances in Computer Science and Technology (ACST)* (2007), ACTA Press, pp. 228–233.

[19] Bonino, D., Castellina, E., Corno, F., Gale, A., Garbo, A., Purdy, K., and Shi, F. A blueprint for integrated eye-controlled environments. *Universal Access in the Information Society 8*, 4 (Oct. 2009), 311–321.

[20] Brenger, B., and Mittelberg, I. Shakes, nods and tilts. motion-capture data profiles of speakers' and listeners' head gestures. In *Proceedings of the 3rd Gesture and Speech in Interaction (GESPIN) Conference* (2015), pp. 43–48.

[21] Bressem, J. Transcription systems for gestures, speech, prosody, postures, and gaze. In *Proceedings of Body - Language - Communication: An International Handbook on Multimodality in Human Interactiong* (2013), vol. 1, pp. 1037–1059.

[22] Brône, G., and Oben, B. Insight interaction: a multimodal and multifocal dialogue corpus. *Language resources and evaluation 49*, 1 (2015), 195–214.

[23] Brône, G., Oben, B., and Goedemé, T. Towards a more effective method for analyzing mobile eye-tracking data: Integrating gaze data with object recognition algorithms. In *Proceedings of the 1st International Workshop on Pervasive Eye Tracking (PETMEI)* (New York, NY, USA, 2011), pp. 53–56.

[24] Buehler, P., Everingham, M., Huttenlocher, D. P., and Zisserman, A. Long term arm and hand tracking for continuous sign language tv broadcasts. In *Proceedings of the British Machine Vision Conference (BMVC)* (2008), BMVA Press, pp. 1105–1114.

[25] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. Brief: Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2010), pp. 778–792.

[26] Carpenter, R. *Movements of the Eyes*. Pion, 1977.

[27] Cave, A. R., Blackler, A. L., Popovic, V., and Kraal, B. J. Examining intuitive navigation in airports. In *Design Research Society Conference 2014* (Umea, Sweden, 2014).

[28] Chang, J. Y. *Nonparametric Gesture Labeling from Multi-modal Data*. Springer International Publishing, Cham, 2015, pp. 503–517.

[29] Cleymans, T. Ontwikkeling van een gebruiksvriendelijke visualisatie-omgeving voor eye- trackeranalyses. In *Master thesis, KU Leuven*. 2014.

[30] Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement 20*, 1 (1960), 37–46.

[31] Crane, H. D. The Purkinje Image Eyetracker, Image Stabilization, and Related Forms of Stimulus Manipulation. In *Visual Science and Engineering: Models and Applications*. 1994, ch. 2, pp. 15–89.

[32] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 886–893.

[33] Davson, H. Preface. In *Physiology of the Eye (Fourth Edition)*, H. Davson, Ed., fourth edition ed. Academic Press, 1980.

[34] De Beugher, S., Brône, G., and Goedemé, T. A semi-automatic annotation tool for unobtrusive gesture analysis. *Language Resources and Evaluation*, Submitted for review.

[35] De Beugher, S., Brône, G., and Goedemé, T. Object recognition and person detection for mobile eye-tracking research: A case study with real-life customer journeys. In *Proceedings of the First International Workshop on Solutions for Automatic Gaze Data Analysis (SAGA)* (Bielefeld, Germany, 2013), pp. 24–26.

[36] De Beugher, S., Brône, G., and Goedemé, T. Automatic analysis of in-the-wild mobile eye-tracking experiments using object, face and person detection. In *Proceedings of the 9th International Conference on Computer Vision Theory And Applications (VISAPP)* (Lisbon, Portugal, 2014), pp. 625–633.

[37] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand detection: A case study on real life mobile eye-tracker data. In *Proceedings of the 10th International Conference on Computer Vision Theory And Applications (VISAPP)* (Berlin, Germany, 2015), pp. 121–129.

[38] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand annotation making human-human interaction analysis fast and accurate. In *Proceedings of the 11th International Conference on Computer Vision Theory And Applications (VISAPP)* (Rome,Italy, 2016), pp. 552–559.

[39] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand annotation of egocentric recordings. In *Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 598. Springer International Publishing, 2016, ch. 18, pp. 338–355.

[40] Dollar, P., Belongie, S., and Perona, P. The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference (BMVC)* (2010), pp. 68.1–68.11.

[41] Dollar, P., Tu, Z., Perona, P., and Belongie, S. Integral channel features. In *Proceedings of the British Machine Vision Conference (BMVC)* (2009), pp. 91.1–91.11.

[42] Dollar, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 4 (2012), 743–761.

[43] Drai-Zerbib, V., Baccino, T., and Bigand, E. Sight-reading expertise: Cross-modality integration investigated using eye tracking. *Psychology of Music 40*, 2 (2012), 216–235.

[44] Dubout, C., and Fleuret, F. Exact acceleration of linear object detectors. In *Proceedings of European Conference on Computer Vision (ECCV)* (2012), vol. 7574, Springer, pp. 301–311.

[45] Duchowski, A. A breadth-first survey of eye-tracking applications. *Behavior Research Methods 34* (2002), 455–470.

[46] Duchowski, A. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[47] Eichner, M., Marin-Jimenez, M., Zisserman, A., and Ferrari, V. 2D articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision 99* (2012), 190—-214.

[48] Escalera, S., Baró, X., Gonzàlez, J., Bautista, M. A., Madadi, M., Reyes, M., Ponce-López, V., Escalante, H. J., Shotton, J., and Guyon, I. *Proceedings of ECCV 2014 Workshops*. Springer International Publishing, Cham, 2015, ch. ChaLearn Looking at People Challenge 2014: Dataset and Results, pp. 459–473.

[49] Evans, K. M., Jacobs, R. A., Tarduno, J. A., and Pelz, J. B. Collecting and analyzing eye-tracking data in outdoor environments. *Journal of Eye Movement Research 5*, 2 (2012).

[50] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. `http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html`.

[51] Felzenszwalb, P., McAllester, D., and Ramanan, D. A discriminatively trained, multiscale, deformable part model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2008).

[52] Ferrari, V., Marin-Jimenez, M., and Zisserman, A. Pose search: Retrieving people using their pose. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2009), pp. 1–8.

[53] Feyaerts, K., Brône, G., and Oben, B. Multimodality in interaction. *Handbook of Cognitive Linguistics* (2016).

[54] Fischler, M. A., and Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communicationsof the ACM 24*, 6 (June 1981), 381–395.

[55] Franchak, J. M., Kretch, K. S., Soska, K. C., Babcock, J. S., and Adolph, K. E. Head-mounted eye-tracking of infants' natural interactions: a new method. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (2010), ACM, pp. 21–27.

[56] Gebre, B. G., Wittenburg, P., and Lenkiewicz, P. Towards automatic gesture stroke detection. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)* (Istanbul, Turkey, may 2012), European Language Resources Association (ELRA).

[57] Girshick, R. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)* (December 2015).

[58] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

[59] Gullberg, M., and Kita, S. Attention to speech-accompanying gestures: Eye movements and information uptake. *Journal of nonverbal behavior 33*, 4 (2009), 251–277.

[60] Hayes, A. F., and Krippendorff, K. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures 1*, 1 (2007), 77–89.

[61] Hayhoe, M. M., and Rothkopf, C. A. Vision in the natural world. *Wiley Interdisciplinary Reviews: Cognitive Science 2*, 2 (2011), 158–166.

[62] Henderson, J. M. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences 7*, 11 (2003), 498 – 504.

[63] Irwin, D. E. *Visual Memory Within and Across Fixations*. Springer New York, New York, NY, 1992, pp. 146–165.

[64] Ishiguro, Y., Mujibiya, A., Miyaki, T., and Rekimoto, J. Aided eyes: Eye activity sensing for daily life. In *Proceedings of the 1st Augmented Human International Conference* (2010), pp. 25:1–25:7.

[65] Itti, L., and Koch, C. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging 10*, 1 (Jan 2001), 161–169.

[66] Jacob, R. J. K. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. In *Advances in Human-Computer Interaction* (1993), pp. 151–190.

[67] Jokinen, K. Non-verbal signals for turn-taking and feedback. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation* (may 2010).

[68] Jokinen, K., Nishida, M., and Yamamoto, S. Eye-gaze experiments for conversation monitoring. In *Proceedings of the 3rd International Universal Communication Symposium (IUCS)* (2009), pp. 303–308.

[69] Jokinen, K., Nishida, M., and Yamamoto, S. On eye-gaze and turn-taking. In *Proceedings of the 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction (EGIHMI)* (2010), pp. 118–123.

[70] Jones, M. J., and Rehg, J. M. Statistical color models with application to skin detection. *International Journal of Computer Vision 46*, 1 (2002), 81–96.

[71] Judd, T., Ehinger, K., Durand, F., and Torralba, A. Learning to predict where humans look. pp. 2106–2113.

[72] Just, M. A., and Carpenter, P. A. A theory of reading: from eye fixations to comprehension. *Psychological review 87*, 4 (1980), 329.

[73] Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D (1960), 35–45.

[74] Kandil, F. I., Rotter, A., and Lappe, M. Car drivers attend to different gaze targets when negotiating closed vs. open bends. *Journal of Vision 10*, 4 (2010), 24.

[75] Kapuscinski, T., Oszust, M., Wysocki, M., and Warchol, D. Recognition of hand gestures observed by depth cameras. *International Journal of Advanced Robotic Systems 12* (2015).

[76] Karlinsky, L., Dinerstein, M., Harari, D., and Ullman, S. The chains model for detecting parts by their context. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 25–32.

[77] Kassner, M., Patera, W., and Bulling, A. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (April 2014).

[78] Ke, Y., and Sukthankar, R. PCA-SIFT: a more distinctive representation for local image descriptors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2004), pp. 506–513.

[79] Kipp, M., Martin, J., Paggio, P., and Heylen, D., Eds. *Multimodal Corpora - From Models of Natural Interaction to Systems and Applications*, vol. 5509. Springer, 2009.

[80] Kolsch, M., and Turk, M. Fast 2d hand tracking with flocks of features and multi-cue integration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2004), pp. 158 – 158.

[81] Krippendorff, K. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement 30*, 1 (1970), 61–70.

[82] Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*. Sage, 2004.

[83] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[84] Kuznetsova, A., Leal-Taixe, L., and Rosenhahn, B. Real-time sign language recognition using a consumer depth camera. In *Procedings of The IEEE International Conference on Computer Vision (ICCV) Workshops* (June 2013).

[85] Land, M., Mennie, N., and Rusted, J. The roles of vision and eye movements in the control of activities of daily living. *Perception 28*, 11 (1999), 1311–1328.

[86] Leutenegger, S., Chli, M., and Siegwart, R. Y. BRISK: Binary Robust Invariant Scalable Keypoints. In *Proceedings of the International Conference on Computer Vision (ICCV)* (Nov 2011), pp. 2548–2555.

[87] Lowe, D. Distinctive image features from scale-invariant keypoints. In *International Journal on Computer Vision IJCV (60)* (2004), pp. 91–110.

[88] Lücking, A., Bergmann, K., Hahn, F., Kopp, S., and Rieser, H. The Bielefeld Speech and Gesture Alignment Corpus (SaGA). In *Proceedings of LREC 2010 Workshop: Multimodal Corpora–Advances in Capturing, Coding and Analyzing Multimodality* (2010), pp. 92–98.

[89] Marcos-Ramiro, A., Pizarro-Perez, D., Marron-Romera, M., Nguyen, L. S., and Gatica-Perez, D. Body communicative cue extraction for conversational analysis. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition* (Apr. 2013).

[90] Marshall, S. P. What the Eyes Reveal: Measuring the Cognitive Workload of Teams Digital Human Modeling. 2009, pp. 265–274.

[91] Matas, J., Chum, O., Urban, M., and Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference (BMVC)* (Cardiff, Wales, 2002), pp. 384–396.

[92] Mathias, M., Benenson, R., Timofte, R., and Van Gool, L. Handling occlusions with franken-classifiers. In *2013 IEEE International Conference on Computer Vision* (2013), pp. 1505–1512.

[93] McNeill, D. *Hand and Mind: What gestures reveal about thought.* University of Chicago Press, Chicago, Illinois, 1992.

[94] Mikolajczyk, K., and Schmid, C. An affine invariant interest point detector. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2002), pp. 128–142.

[95] Mikolajczyk, K., and Schmid, C. A performance evaluation of local descriptors. *PAMI 27*, 10 (2005), 1615–1630.

[96] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. A comparison of affine region detectors. *International Journal on Computer Vision* (2005).

[97] Miksik, O., and Mikolajczyk, K. Evaluation of local detectors and descriptors for fast feature matching. In *Proceedings of International Conference on Pattern Recognition (ICPR)* (2012), pp. 2681–2684.

[98] Mittal, A., Zisserman, A., and Torr, P. Hand detection using multiple proposals. In *Proceedings of the British Machine Vision Conference (BMVC)* (2011), BMVA Press, pp. 75.1–75.11.

[99] Monnier, C., German, S., and Ost, A. *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I.* 2015, ch. A Multi-scale Boosted Detector for Efficient and Robust Gesture Recognition.

[100] Morel, J.-M., and Yu, G. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences 2*, 2 (Apr. 2009), 438–469.

[101] Neuendorf, K. *The Content Analysis Guidebook*. SAGE Publications, 2002.

[102] Neverova, N., Wolf, C., W.Taylor, G., and Nebout, F. Multi-scale deep learning for gesture detection and localization. In *Proceedings of ECCV ChaLearn Workshop on Looking at People* (Sept. 2014).

[103] Pang, Y., Li, W., Yuan, Y., and Pan, J. Fully affine invariant SURF for image matching. *Neurocomputing 85*, 0 (2012), 6 – 10.

[104] Parks, D., Borji, A., and Itti, L. Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes. *Vision Research 116B* (2015), 113–126.

[105] Peng, X., Wang, L., Cai, Z., and Qiao, Y. *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I.* Springer International Publishing, Cham, 2015, ch. Action and Gesture Temporal Spotting with Super Vector Representation, pp. 518–527.

[106] Popping, R. *On Agreement Indices for Nominal Data*. Palgrave Macmillan UK, London, 1988.

[107] Raheja, J. L., Chaudhary, A., and Singal, K. Tracking of fingertips and centers of palm using KINECT. In *In Proceedings of Third International Conference on Computational Intelligence, Modelling Simulation* (Sept 2011), pp. 248–252.

[108] Rautaray, S. S., and Agrawal, A. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review 43*, 1 (2012), 1–54.

[109] Rayner, K. Eye movements and attention in reading, scene perception, and visual search. *The Quarterly Journal of Experimental Psychology 62*, 8 (2009), 1457–1506.

[110] Ren, Z., Yuan, J., Meng, J., and Zhang, Z. Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia 15*, 5 (Aug 2013), 1110–1120.

[111] Richardson, D. C., and Spivey, M. J. Eye tracking: Characteristics and methods. *Encyclopedia of biomaterials and biomedical engineering* (2004), 568–572.

[112] Rosten, E., and Drummond, T. Fusing points and lines for high performance tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2005), pp. 1508–1515.

[113] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision (ICCV)* (Nov 2011), pp. 2564–2571.

[114] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252.

[115] Schmid, C., Mohr, R., and Bauckhage, C. Local grey-value invariants for image retrieval. *International Journal on Pattern Analysis an Machine Intelligence 19*, 5 (1997), 872–877.

[116] Schreer, O., and Masneri, S. Automatic video analysis for annotation of human body motion in humanities research. In *International Workshop on Multimodal Corpora in conjunction with 9th edition of the Language Resources and Evaluation Conference (LREC)* (2014), pp. 29–32.

[117] Schwarzkopf, S., von Stülpnagel, R., Büchner, S. J., Konieczny, L., Kallert, G., and Hölscher, C. What lab eye-tracking tells us about wayfinding. a comparison of stationary and mobile eye-tracking in a large building scenario. In *1st Intl. workshop on eye tracking for spatial research, ET4S* (2013).

[118] Scott, N., Green, C., and Fairley, S. Investigation of the use of eye tracking to examine tourism advertising effectiveness. *Current Issues in Tourism 19*, 7 (2016), 634–642.

[119] Scott, W. A. Reliability of content analysis: The case of nominal scale coding. *The Public Opinion Quarterly 19*, 3 (1955), 321–325.

[120] Shan, C., Tan, T., and Wei, Y. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognition 40*, 7 (2007), 1958 – 1970.

[121] Shebilske, W. L., and Fisher, D. F. Understanding extended discourse through the eyes: How and why. 303–314.

[122] Smedt, F. D., Beeck, K. V., Tuytelaars, T., and Goedeme, T. The combinator: Optimal combination of multiple pedestrian detectors. In *Proceedings of International Conference on Pattern Recognition (ICPR)* (Aug 2014), pp. 3522–3527.

[123] Spruyt, V., Ledda, A., and Philips, W. Real-time, long-term hand tracking with unsupervised initialization. In *Proceedings of IEEE International Conference on Image Processing (ICIP)* (2013), pp. 3730–3734.

[124] Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., and Tröster, G. Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *Proceedings of the International Symposium on Wearable Computers ISWC* (2006), pp. 97–104.

[125] Toyama, T., Kieninger, T., Shafait, F., and Dengel, A. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA)* (2012), pp. 91–98.

[126] Tsotsos, J. K., Eckstein, M. P., and Landy, M. S. Computational models of visual attention. *Vision Research 116, Part B* (2015), 93 – 94. Computational Models of Visual Attention.

[127] Tuytelaars, T., and Mikolajczyk, K. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision 3*, 3 (July 2008), 177–280.

[128] Tuytelaars, T., and Van Gool, L. Wide baseline stereo based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference (BMVC)* (Bristol, UK, 2000), pp. 412–422.

[129] Tuytelaars, T., Van Gool, L., D'haene, L., and Koch, R. Matching of affinely invariant regions for visual servoing. In *Proceedings of The International Conference on Robotics and Automation (ICRA)* (1999), pp. 1601–1606.

[130] Van Beeck, K. The automatic blind spot camera: hard real-time detection of moving objects from a moving camera. In *PhD dissertation, KU Leuven.* 2016.

[131] van Gompel, R. *Eye Movements: A Window on Mind and Brain.* Elsevier Science, May 2007.

[132] Vandemoortele, S., De Beugher, S., Brône, G., Feyaerts, K., Goedemé, T., De Baets, T., and Vervliet, S. Into the wild – musical communication in ensemble playing. discerning mutual and solitary gaze events in musical duos using mobile eye-tracking. In *Proceedings of the 2nd International Workshop on Solutions for Automatic Gaze Data Analysis (SAGA)* (Bielefeld, Germany, 2015).

[133] Vansteenkiste, P., Cardon, G., Philippaerts, R., and Lenoir, M. High quality bicycle tracks result in more efficient visual search patterns during cycling. In *Scandinavian Workshop on Applied Eye Tracking, Abstracts* (2012), pp. 35–35.

[134] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2001), vol. 1, pp. I–511–I–518 vol.1.

[135] Viola, P., Jones, M. J., and Snow, D. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of International Conference on Computer Vision (ICCV)* (2003), pp. 734–741 vol.2.

[136] Wang, R. Y., and Popović, J. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics 28*, 3 (2009).

[137] Wills, A. J., Lavric, A., Croft, G. S., and Hodgson, T. L. Predictive learning, prediction errors, and attention: Evidence from event-related potentials and eye tracking. *Journal of Cognitive Neuroscience 19*, 5 (May 2007), 843–854.

[138] Yang, Y., and Ramanan, D. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), IEEE, pp. 1385–1392.

[139] Ye, Z., Li, Y., Fathi, A., Han, Y., Rozga, A., Abowd, G. D., and Rehg, J. M. Detecting eye contact using wearable eye-tracking glasses. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (2012), pp. 699–704.

[140] Yin, Y., and Davis, R. Gesture spotting and recognition using salience detection and concatenated hidden markov models. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction (ICMI)* (New York, NY, USA, 2013), ACM, pp. 489–494.

[141] Young, L. R., and Sheena, D. Survey of eye movement recording methods. *Behavior Research Methods & Instrumentation 7*, 5 (1975), 397–429.

[142] Yun, K., Peng, Y., Samaras, D., Zelinsky, G. J., and Berg, T. L. Studying relationships between human gaze, description, and computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013), pp. 739–746.

[143] Zhang, Z., Conly, C., and Athitsos, V. Hand detection on sign language videos. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA)* (2014), ACM, pp. 26:1–26:5.

List of publications

International conference publications

[1] De Beugher, S., Brône, G., and Goedemé, T. Object recognition and person detection for mobile eye-tracking research: A case study with real-life customer journeys. In *Proceedings of the First International Workshop on Solutions for Automatic Gaze Data Analysis (SAGA)* (Bielefeld, Germany, 2013), pp. 24–26.

[2] De Beugher, S., Brône, G., and Goedemé, T. Automatic analysis of in-the-wild mobile eye-tracking experiments using object, face and person detection. In *Proceedings of the 9th International Conference on Computer Vision Theory And Applications (VISAPP)* (Lisbon, Portugal, 2014), pp. 625–633.

[3] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic annotation of eye-tracking recordings in terms of human torso, face and hands. In *Proceedings of the 2nd International Workshop on Solutions for Automatic Gaze Data Analysis (SAGA)* (Bielefeld, Germany, 2015), pp. 7–10.

[4] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand detection: A case study on real life mobile eye-tracker data. In *Proceedings of the 10th International Conference on Computer Vision Theory And Applications (VISAPP)* (Berlin, Germany, 2015), pp. 121–129.

[5] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand annotation making human-human interaction analysis fast and accurate. In *Proceedings of the 11th International Conference on Computer Vision Theory And Applications (VISAPP)* (Rome,Italy, 2016), pp. 552–559.

[6] De Beugher, S., Ichiche, Y., Brône, G., and Goedemé, T. Automatic analysis of eye-tracking data using object detection algorithms.

In *Proceedings of the ACM conference on ubiquitous computing: workshop session: perasive eye traking and mobile eye-based interaction (PETMEI)* (Pittsburgh, USA, 2012).

[7] Struyf, L., De Beugher, S., Van Uytsel, D. H., Kanters, F., and Goedemé, T. The battle of the giants: a case study of gpu vs fpga optimisation for real-time image processing. In *Proceedings of the 4th International conference on pervasive and embedded computing and communication systems (PECCS)* (Lisbon, Portugal, 2014), pp. 112–119.

[8] Vandemoortele, S., De Beugher, S., Brône, G., Feyaerts, K., Goedemé, T., De Baets, T., and Vervliet, S. Into the wild – musical communication in ensemble playing. discerning mutual and solitary gaze events in musical duos using mobile eye-tracking. In *Proceedings of the 2nd International Workshop on Solutions for Automatic Gaze Data Analysis (SAGA)* (Bielefeld, Germany, 2015).

Book chapter publications

[1] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand annotation of egocentric recordings. In *Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 598 of *Communications in Computer and Information Science*. Springer International Publishing, 2016, ch. 18, pp. 338–355.

[2] De Beugher, S., Brône, G., and Goedemé, T. Semi-automatic hand annotation of egocentric recordings. In *Eye-tracking in interaction*, vol. 1 of *Advances in Interaction Studies.*, 2016, p. To be published.

Journal publications

[1] De Beugher, S., Brône, G., and Goedemé, T. A semi-automatic annotation tool for unobtrusive gesture analysis. *Language Resources and Evaluation*, Submitted for review.

Abstract publications

[1] De Beugher, S., Brône, G., and Goedemé, T. Towards a system for the semi-automatic annotation of eye gaze data in face-to-face interactions.

In *International Workshop: Mapping Multimodal Dialogue (MaMuD)* (Lille, France, 2015).

[2] De Beugher, s., Tuytelaars, T., Brône, G., and Goedemé, T. Computer vision technique for automatic analysis of eye-tracking data. In *Research Day FIIW-ESAT/CW* (Leuven, Belgium, 2015).

[3] De Beugher, S., Tuytelaars, T., and Goedemé, T. Automatic analysis of in-the-wild mobile eye-tracking experiments using image processing techniques. In *European conference on the use of modern information and communication technologies (ECUMICT)* (Ghent, Belgium, 2014).

[4] Vandemoortele, S., De Beugher, S., Brône, G., Feyaerts, K., Goedemé, T., De Baets, T., and Vervliet, S. Communicative gaze behaviour in ensemble playing. a pilot study using mobile eye-tracking. In *Specialist Course: Expressive interaction with music, humans, and machines location* (Ghent, Belgium, 2015).

[5] Vandemoortele, S., De Beugher, S., Brône, G., Feyaerts, K., Goedemé, T., De Baets, T., and Vervliet, S. Studying musicians' gaze behaviour in the light of synchronisation issues in ensemble playing. In *Making Time in Music* (Oxford, United Kingdom, 2016).

[6] De Beugher, S., Brône, G., and Goedemé, T. Automatic analysis of in-the-wild mobile eye-tracking experiments. In *The first International Workshop on Egocentric Perception, Interaction and Computing (EPIC)* (Amsterdam, The Netherlands, 2016).

Curriculum Vitae

Stijn De Beugher was born on the 26th of July 1988 in Duffel, Belgium. He obtained his Master in Industrial Sciences - Electronics-ICT at the KU Leuven Technology Campus De Nayer in 2011. Because of his interest in applied electronics and his eagerness to expand his knowledge about this research field, he decided to start his career as a researcher in the EAVISE research group, under the supervision of Prof. dr. ir. Toon Goedemé. The main focus of EAVISE is to transfer and apply computer vision techniques into real-life and challenging applications, which are mostly industry-driven. During the first year as a researcher, in which he optimised a real-time on-loom textile camera inspection system, his passion for computer vision was sparked. He started his doctoral research in 2012 under the supervision of Prof. dr. ir. Toon Goedemé. In his PhD project, he focused on the application of computer vision algorithms for the analysis of real-life mobile eye-tracking recordings. In the first part of his PhD, the focus was on commercially driven applications of mobile eye-tracking like market research, whereas later on, this focus shifted towards more academic oriented applications, such as human-human interaction analysis.

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING
EMBEDDED ARTIFICIALLY INTELLIGENT VISION ENGINEERING (EAVISE)
Jan De Nayerlaan 5
B-2860 Sint-Katelijne-Waver
stijn.debeugher@kuleuven.be
http://www.eavise.be